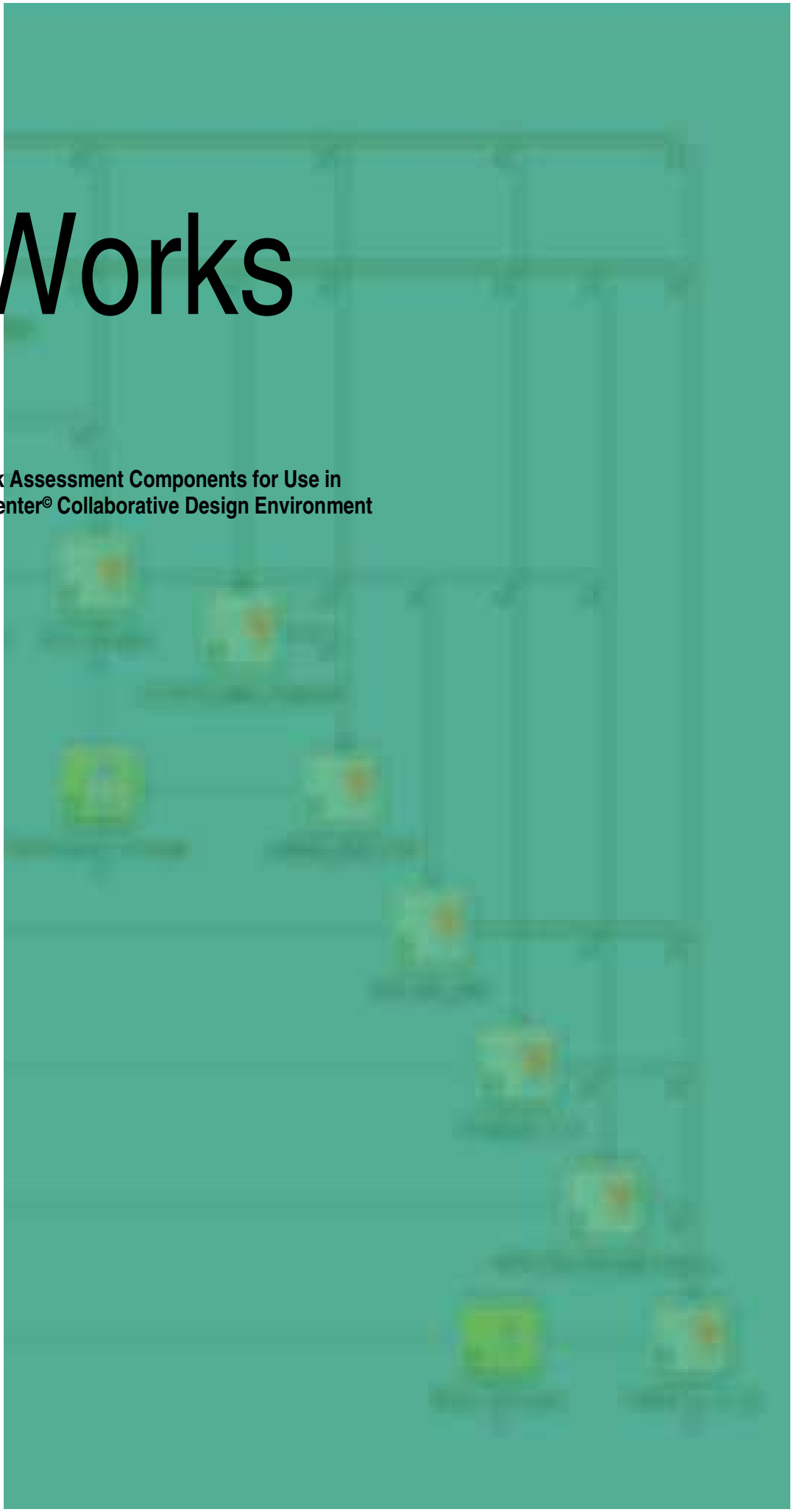
ProbWorks

version 1.1

A Suite of Probability and Risk Assessment Components for Use in Phoenix Integration's ModelCenter© Collaborative Design Environment



PiBlue

ProbWorks Version 1.1
User Manual

Copyright ©2004, Pi Blue Software, Inc., Alpharetta, GA, USA

All rights reserved. Written, designed, and published in the United States of America.

This manual, and the software described within it, are furnished under license and may only be used or copied in accordance with the terms of the license agreement. Such software is subject to the terms and conditions of the enclosed license agreement. Information in this document is provided for informational purposes only, is subject to change without notice, and does not represent a commitment as to the merchantability or fitness for a particular purpose by Pi Blue Software, Inc..

No part of this manual may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying and recording, for any purpose without the express written permission of Pi Blue Software, Inc.

To purchase additional copies of this document, contact the Technical Products Department at the address below:

Pi Blue Software, Inc.
170 Park Creek Drive
Alpharetta, GA, USA 30005

Web: www.piblue.com
E-mail: info@piblue.com

First printing, January 30, 2004.
User Manual Version 1.1

Other product names mentioned herein may be trademarks and/or registered trademarks of the respective holders.

ModelCenter© is a registered trademark of Phoenix Integration, Inc.
Analysis Server© is a registered trademark of Phoenix Integration, Inc.

ProbWorks Version 1.1
SOFTWARE LICENSE AGREEMENT AND WARRANTY STATEMENT
(Single CPU/End-User License)

PI BLUE SOFTWARE, INC. IS WILLING TO LICENSE THE SOFTWARE ONLY UPON THE CONDITION THAT YOU ACCEPT ALL OF THE TERMS CONTAINED IN THIS LICENSE AGREEMENT. PLEASE READ THE TERMS CAREFULLY BEFORE INSTALLING THE SOFTWARE, AS INSTALLING THE SOFTWARE WILL INDICATE YOUR AGREEMENT WITH THEM. IF YOU DO NOT AGREE WITH THESE TERMS, THEN PI BLUE SOFTWARE IS UNWILLING TO LICENSE THE SOFTWARE TO YOU, IN WHICH EVENT YOU SHOULD RETURN THE SOFTWARE TO THE PLACE OF PURCHASE AND RECEIVE A FULL REFUND.

1. The Software. The accompanying computer programs, data compilation(s), and documentation are referred to herein as the "Software".

2. Perpetual Term. The term of the license granted herein shall be perpetual unless terminated by written notice by You for convenience or terminated by either party for material breach. Immediately upon termination of this license for any reason, You shall return to Pi Blue Software all copies of the Software and documentation.

3. License Grant (Single, Standalone CPU). You are granted non-exclusive rights to install and use the Software only on a single computer. You may not install the Software on a network, transmit the Software electronically from one computer to another or over a network, or otherwise share the Software with another user. You may copy the Software for archival purposes, provided that any copy must contain the original Software's proprietary notices in unaltered form.

4. Restrictions. You may not: (i) permit others to use the Software; (ii) modify or translate the Software; (iii) reverse engineer, decompile, or disassemble the Software, except to the extent this restriction is expressly prohibited by applicable law; (iv) create derivative works based on the Software; (v) merge the Software with another product; (vi) copy the Software, except as expressly provided above; or (vii) remove or obscure any proprietary rights notices or labels on the Software.

5. Transfers. You may not transfer the Software or any rights under this Agreement without the prior written consent of Pi Blue Software, which consent shall not be unreasonably withheld. A condition to any transfer or assignment shall be that the recipient agrees to the terms of this Agreement. Any attempted transfer or assignment in violation of this provision shall be null and void.

6. Ownership. Pi Blue Software and its suppliers own the Software and all intellectual property rights embodied therein, including copyrights and valuable trade secrets embodied in the Software's design and coding methodology. The Software is protected by United States copyright laws and international treaty provisions. This Agreement provides You only a limited use license, and no ownership of any intellectual property.

LIMITED WARRANTY STATEMENT; LIMITATION OF LIABILITY. Pi Blue Software warrants only to You that the Software shall perform substantially in accordance with accompanying documentation under normal use for a period of ninety (90) days from the purchase date. The entire and exclusive liability and remedy for breach of this Limited Warranty shall be, at Pi Blue Software's option, either (i) return of the list price of the Software, or (ii) replacement of defective Software and/or documentation provided the Software and/or documentation is returned to Pi Blue Software with a copy of your purchase confirmation. PI BLUE SOFTWARE AND ITS SUPPLIERS AND RESELLERS SPECIFICALLY DISCLAIM THE IMPLIED WARRANTIES OF TITLE, NON-INFRINGEMENT, MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, SYSTEM INTEGRATION, AND DATA ACCURACY. THERE IS NO WARRANTY OR GUARANTEE THAT THE OPERATION OF THE SOFTWARE WILL BE UNINTERRUPTED, ERROR-FREE, OR VIRUS-FREE, OR THAT THE SOFTWARE WILL MEET ANY PARTICULAR CRITERIA OF PERFORMANCE, QUALITY, ACCURACY, PURPOSE, OR NEED, EXCEPT AS EXPRESSLY PROVIDED IN THE LIMITED WARRANTY. THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS AGREEMENT. NO USE OF THE SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER THIS DISCLAIMER. No action for the above Limited Warranty may be commenced after one (1) year following the expiration date of the warranty. If implied warranties may not be disclaimed under applicable law, then ANY IMPLIED WARRANTIES ARE LIMITED IN DURATION TO ninety (90) days FROM THE PURCHASE DATE. Some states do not allow limitations on how long an implied warranty may last, so the above limitations may not apply to You. This warranty gives you specific rights, and You may have other rights which vary from state to state.

INDEPENDENT OF THE FORGOING PROVISIONS, IN NO EVENT AND UNDER NO LEGAL THEORY, INCLUDING WITHOUT LIMITATION, TORT, CONTRACT, OR STRICT PRODUCTS LIABILITY, SHALL PI BLUE SOFTWARE OR ANY OF ITS SUPPLIERS BE LIABLE TO YOU OR ANY OTHER PERSON FOR ANY INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES OF ANY KIND, INCLUDING WITHOUT LIMITATION, DAMAGES FOR LOSS OF GOODWILL, WORK STOPPAGE, COMPUTER MALFUNCTION, OR ANY OTHER KIND OF COMMERCIAL DAMAGE, EVEN IF PI BLUE SOFTWARE HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. THIS LIMITATION SHALL NOT APPLY TO LIABILITY FOR DEATH OR PERSONAL INJURY TO THE EXTENT PROHIBITED BY APPLICABLE LAW.

IN NO EVENT SHALL PI BLUE SOFTWARE'S LIABILITY FOR ACTUAL DAMAGES FOR ANY CAUSE WHATSOEVER, AND REGARDLESS OF THE FORM OF ACTION, EXCEED THE AMOUNT OF THE PURCHASE PRICE PAID FOR THE SOFTWARE LICENSE.

EXPORT PROHIBITION. The Software is intended for use in the United States only. By your acceptance of the Software as indicated above, You agree not to transmit or otherwise export the Software or underlying information or technology from the United States.

MISCELLANEOUS. This Agreement constitutes the entire understanding of the parties with respect to the subject matter of this Agreement and merges all prior communications, representations, and agreements. This Agreement may be modified only by a written agreement signed by the parties. If any provision of this Agreement is held to be unenforceable for any reason, such provision shall be reformed only to the extent necessary to make it enforceable. This Agreement shall be construed under the laws of the State of Georgia, excluding rules regarding conflicts of law. This license is written in English and English is its controlling language.

U.S. GOVERNMENT END USERS. The Software and documentation are provided with RESTRICTED RIGHTS. Use, duplication, or disclosure by the United States Government is subject to restrictions as set forth in subparagraph (c)(1) and (2) of the Commercial Computer Software-Restricted Rights at 48 CFR 52.227-19, as applicable. Manufacturer is Pi Blue Software, Inc., 170 Park Creek Drive, Alpharetta, GA, USA 30005.

ProbWorks Version 1.1
SOFTWARE LICENSE AGREEMENT AND WARRANTY STATEMENT
(Division-Wide End-User License)

PI BLUE SOFTWARE, INC. IS WILLING TO LICENSE THE SOFTWARE ONLY UPON THE CONDITION THAT YOU ACCEPT ALL OF THE TERMS CONTAINED IN THIS LICENSE AGREEMENT. PLEASE READ THE TERMS CAREFULLY BEFORE INSTALLING THE SOFTWARE, AS INSTALLING THE SOFTWARE WILL INDICATE YOUR AGREEMENT WITH THEM. IF YOU DO NOT AGREE WITH THESE TERMS, THEN PI BLUE SOFTWARE IS UNWILLING TO LICENSE THE SOFTWARE TO YOU, IN WHICH EVENT YOU SHOULD RETURN THE SOFTWARE TO THE PLACE OF PURCHASE AND RECEIVE A FULL REFUND.

1. The Software. The accompanying computer programs, data compilation(s), and documentation are referred to herein as the "Software".

2. Perpetual Term. The term of the license granted herein shall be perpetual unless terminated by written notice by You for convenience or terminated by either party for material breach. Immediately upon termination of this license for any reason, You shall return to Pi Blue Software all copies of the Software and documentation.

3. License Grant (Division-Wide License). You are granted non-exclusive rights to install and use the Software on a network and/or on an unlimited number of computers with an unlimited number of users, provided that all such computers and users are permanently assigned to the division of your organization specified on the accompanying invoice. You may not install or use the Software outside of such division of your organization. You may copy the Software for archival purposes, provided that any copy must contain the original Software's proprietary notices in unaltered form.

4. Restrictions. You may not: (i) permit others to use the Software, except as expressly provided above; (ii) modify or translate the Software; (iii) reverse engineer, decompile, or disassemble the Software, except to the extent this restriction is expressly prohibited by applicable law; (iv) create derivative works based on the Software; (v) merge the Software with another product; (vi) copy the Software, except as expressly provided above; or (vii) remove or obscure any proprietary rights notices or labels on the Software.

5. Transfers. You may not transfer the Software or any rights under this Agreement without the prior written consent of Pi Blue Software, which consent shall not be unreasonably withheld. A condition to any transfer or assignment shall be that the recipient agrees to the terms of this Agreement. Any attempted transfer or assignment in violation of this provision shall be null and void.

6. Ownership. Pi Blue Software and its suppliers own the Software and all intellectual property rights embodied therein, including copyrights and valuable trade secrets embodied in the Software's design and coding methodology. The Software is protected by United States copyright laws and international treaty provisions. This Agreement provides You only a limited use license, and no ownership of any intellectual property.

LIMITED WARRANTY STATEMENT; LIMITATION OF LIABILITY. Pi Blue Software warrants only to You that the Software shall perform substantially in accordance with accompanying documentation under normal use for a period of ninety (90) days from the purchase date. The entire and exclusive liability and remedy for breach of this Limited Warranty shall be, at Pi Blue Software's option, either (i) return of the list price of the Software, or (ii) replacement of defective Software and/or documentation provided the Software and/or documentation is returned to Pi Blue Software with a copy of your purchase confirmation. PI BLUE SOFTWARE AND ITS SUPPLIERS AND RESELLERS SPECIFICALLY DISCLAIM THE IMPLIED WARRANTIES OF TITLE, NON-INFRINGEMENT, MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, SYSTEM INTEGRATION, AND DATA ACCURACY. THERE IS NO WARRANTY OR GUARANTEE THAT THE OPERATION OF THE SOFTWARE WILL BE UNINTERRUPTED, ERROR-FREE, OR VIRUS-FREE, OR THAT THE SOFTWARE WILL MEET ANY PARTICULAR CRITERIA OF PERFORMANCE, QUALITY, ACCURACY, PURPOSE, OR NEED, EXCEPT AS EXPRESSLY PROVIDED IN THE LIMITED WARRANTY. THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS AGREEMENT. NO USE OF THE SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER THIS DISCLAIMER. No action for the above Limited Warranty may be commenced after one (1) year following the expiration date of the warranty. If implied warranties may not be disclaimed under applicable law, then ANY IMPLIED WARRANTIES ARE LIMITED IN DURATION TO ninety (90) days FROM THE PURCHASE DATE. Some states do not allow limitations on how long an implied warranty may last, so the above limitations may not apply to You. This warranty gives you specific rights, and You may have other rights which vary from state to state.

INDEPENDENT OF THE FORGOING PROVISIONS, IN NO EVENT AND UNDER NO LEGAL THEORY, INCLUDING WITHOUT LIMITATION, TORT, CONTRACT, OR STRICT PRODUCTS LIABILITY, SHALL PI BLUE SOFTWARE OR ANY OF ITS SUPPLIERS BE LIABLE TO YOU OR ANY OTHER PERSON FOR ANY INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES OF ANY KIND, INCLUDING WITHOUT LIMITATION, DAMAGES FOR LOSS OF GOODWILL, WORK STOPPAGE, COMPUTER MALFUNCTION, OR ANY OTHER KIND OF COMMERCIAL DAMAGE, EVEN IF PI BLUE SOFTWARE HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. THIS LIMITATION SHALL NOT APPLY TO LIABILITY FOR DEATH OR PERSONAL INJURY TO THE EXTENT PROHIBITED BY APPLICABLE LAW.

IN NO EVENT SHALL PI BLUE SOFTWARE'S LIABILITY FOR ACTUAL DAMAGES FOR ANY CAUSE WHATSOEVER, AND REGARDLESS OF THE FORM OF ACTION, EXCEED THE AMOUNT OF THE PURCHASE PRICE PAID FOR THE SOFTWARE LICENSE.

EXPORT PROHIBITION. The Software is intended for use in the United States only. By your acceptance of the Software as indicated above, You agree not to transmit or otherwise export the Software or underlying information or technology from the United States.

MISCELLANEOUS. This Agreement constitutes the entire understanding of the parties with respect to the subject matter of this Agreement and merges all prior communications, representations, and agreements. This Agreement may be modified only by a written agreement signed by the parties. If any provision of this Agreement is held to be unenforceable for any reason, such provision shall be reformed only to the extent necessary to make it enforceable. This Agreement shall be construed under the laws of the State of Georgia, excluding rules regarding conflicts of law. This license is written in English and English is its controlling language.

U.S. GOVERNMENT END USERS. The Software and documentation are provided with RESTRICTED RIGHTS. Use, duplication, or disclosure by the United States Government is subject to restrictions as set forth in subparagraph (c)(1) and (2) of the Commercial Computer Software-Restricted Rights at 48 CFR 52.227-19, as applicable. Manufacturer is Pi Blue Software, Inc., 170 Park Creek Drive, Alpharetta, GA, USA 30005.

CONTENTS

Acknowledgements 13

Introduction 15

Product Overview 15

Manual Organization 16

Installation Files 16

Registration 18

Additional Resources 19

Technical Support 19

Using Online Support 19

Consulting Referral Services 19

Conventions Used in this Manual 19

Chapter 1: General Functionality and Setup 21

Basic Functionality 21

Components of the ProbWorks Suite 21

General Component Operation 22

ProbWorks Interface 23

ProbWorks Installation and Sample Usage 25

Chapter 2: ProbWorks – Monte Carlo 31

Overview 31

Mathematical Foundation 31

Input-Output Variables 32

Sample Usage 37

Simple Analysis: Analytical Function 37

Complex Analysis: Analytical Function 40

Chapter 3: ProbWorks – DPOMD 43

Overview 43

Mathematical Foundation 43

Input-Output Variables 44

Sample Usage 47

Simple Analysis: Analytical Function 47

Complex Analysis: Analytical Function 50

Chapter 4: ProbWorks – RSE Generator 53
Overview 53
Mathematical Foundation 53
Input-Output Variables 54
Sample Usage 57
Simple Analysis: Analytical Function 57
Complex Analysis: Analytical Function 60

Chapter 5: ProbWorks – Pareto Analysis 65
Overview 65
Mathematical Foundation 65
Input-Output Variables 66
Sample Usage 68
Simple Analysis: Analytical Function 68
Complex Analysis: Analytical Function 71

Chapter 6: ProbWorks – Graph Plug-in Utility 75
Overview 75
Component Operation 75
Sample Usage 78
Monte Carlo Analysis: Histogram 78
Pareto Analysis: Pareto Chart 79

Chapter 7: Complete Probabilistic Analysis 81
Overview 81
Probabilistic Analysis Process 81
Probabilistic Analysis Example 82

REFERENCE LIST 85

LIST OF ACRONYMS 87

LIST OF FIGURES 89

LIST OF TABLES 91

INDEX _____ **93**

PRODUCTION NOTES_____ **97**

ACKNOWLEDGEMENTS

ORIGIN

ProbWorks is a complementary but separate suite of software for Phoenix Integration's ModelCenter© and Analysis Server©. The software was conceived as part of a set of packaged solutions to be used for the ModelCenter© collaborative design environment, specific to probabilistic analysis. Version 1 of ProbWorks went into production during April 2002 with version 1.1 completed by October 2002.

ACKNOWLEDGEMENTS

Appreciation is expressed to Phoenix Integration, Inc. for providing encouragement for this initiative as well as general technical help on ModelCenter© and Analysis Server© throughout the development of this product. Specific acknowledgements are expressed to Brett Malone, Scott Woyak, Nathan Sharp, Doug Anderson, Mike Haisma, Kevin Slocum, and Peg Warren from Phoenix Integration, Inc.

INTRODUCTION

PRODUCT OVERVIEW

Pi Blue Software introduces a suite of uncertainty and sensitivity analysis tools for use with Phoenix Integration's ModelCenter© collaborative design environment. Entitled ProbWorks, this suite consists of four tools to help employ uncertainty analysis techniques and a graphical analysis utility, each implemented as a Java-based component which can function on any platform running Phoenix Integration's ModelCenter© and Analysis Server©. The ProbWorks suite is aimed at users who must treat uncertainty and risk in their product designs. The direct Monte Carlo driver and the faster DPOMD approximation driver propagate the influences of uncertainty in input parameters through a user's disciplinary tool to assess the likely outcomes in terms of statistical parameters such as mean, standard deviation, certainty level, and skewness. Supporting tools for the generation of fast-acting polynomial response surface equations (RSEs) and Pareto sensitivity analysis for variable screening are included in the package (see Table 1).

Table 1. Overview of ProbWorks ModelCenter© Components

COMPONENT NAME	CAPABILITY
ProbWorks_MonteCarlo	Performs Monte Carlo uncertainty simulation using random variables by placing distributions (normal, triangular, Weibull, etc.) on inputs. Generates output statistics for the forecast variables (average, mean, certainty level, etc.) even as simulation is running.
ProbWorks_DPOMD	Implements the Discrete Probability Optimal Matching Distribution (DPOMD) technique that serves as an efficient alternative to direct Monte Carlo simulation for certain classes of problems. Allows estimation of a probabilistic output distribution with a small number of runs.
ProbWorks_ParetoSensitivity	Determines the contribution or sensitivity of each selected input with respect to each selected output with appropriate ranking of contribution.
ProbWorks_RSEgenerator	Produces polynomial regression equations to approximate more complex or time-consuming components enabling faster execution of probabilistic techniques such as Monte Carlo. Generates output statistics on goodness of fit to selected data. Enables subsequent use of regression coefficients.
ProbWorks_GraphPlugin	ModelCenter© plug-in which generates histograms and Pareto plots from other ProbWorks component outputs

Each of the algorithms in the suite is coded as a Java component and can be dragged-and-dropped into a ModelCenter© model file from a local or remote Analysis Server© (these are

generally referred to as Analysis Server© components). The Graph PlugIn utility is installed as a ModelCenter© plug-in.

MANUAL ORGANIZATION

This manual provides an overview of the ProbWorks suite including detailed descriptions of each component along with examples case studies in the use of each on simple and complex uncertainty analysis problems.

The manual assumes the user possess a relatively rudimentary knowledge of uncertainty analysis and some familiarity with the ModelCenter© collaborative design environment. The user should recognize basic uncertainty problems and distributions and be able to install new "analyses" on their computers running Analysis Server©. Knowledge of the particular uncertainty analysis method employed in each component is not assumed. Knowledge of the Java or VBScript programming languages is also not assumed.

Chapter 1 offers a top-level overview of each component of the suite including description of each component, general operation and interface within a ModelCenter© environment, methods for installation of the suite, and a general treatment on the mathematical foundation of the probabilistic methods (subsequent chapters include more detail on each particular analysis).

Chapters 2 though 6 separately address each component within the ProbWorks suite. Each of these chapters includes an overview of the component, further mathematical foundation of the method employed, listing of the input / output variables for each component, and sample usage of each component on a relatively simple and complex analysis.

Chapter 7 showcases a complete probabilistic uncertainty analysis using all of the ProbWorks components. A process is described for performing an uncertainty analysis, detailing each step. An example of this process using the ProbWorks components is given.

INSTALLATION FILES

The files on the ProbWorks installation CD include both components (i.e. Java .jar files) to be placed on the user's Analysis Server© as well as various ModelCenter© .pxc "models" that contain both the analysis components and sample case studies. In addition an Adobe© PDF version of the ProbWorks manual is also included along with associated single and group license agreements (both are included) that apply for the license granted the user by Pi Blue Software, Inc.

Each of the examples related in Chapters 2 through 7 are included on the installation CD. This specifically refers to the ModelCenter© .pxc "models" included on the CD. The user will have to reconfigure each of these .pxc files (once opened in ModelCenter©) to point to the Analysis Server© hosting these components. Each of the analytical problems used as examples are coded in VBScript and available in their appropriate ModelCenter© .pxc file on the installation CD. Chapter 1 includes specific information for installation of these files to be accessible to the user as Analysis Server© components.

A separate readme.txt file is also included that lists the files on the CD and separate contact information.

Table 2 lists the specific folders and files on the ProbWorks installation CD.

Table 2. ProbWorks Installation CD Overview

PROBWORKS INSTALLATION CD FOLDER	FILES
ProbWorks Java Components\	ProbWorks_MonteCarlo.jar ProbWorks_DPOMD.jar ProbWorks_ParetoSensitivity.jar ProbWorks_RSEgenerator.jar ProbWorks_GraphPlugin.jar RSEfile.scriptWrapper SEI_logo_small.bmp
ProbWorks Example Problems\ Generic Example Scripts Used\	process_example.pxc simple_example.pxc
ProbWorks Example Problems\ Monte Carlo Examples\	MonteCarlo_simple_example.pxc MonteCarlo_complex_example.pxc
ProbWorks Example Problems\ DPOMD Examples\	DPOMD_simple_example.pxc DPOMD_complex_example.pxc
ProbWorks Example Problems\ RSE Generator Examples\	RSEgen_simple_example.pxc RSEgen_complex_example.pxc
ProbWorks Example Problems\ Pareto Analysis Examples\	Pareto_simple_example.pxc Pareto_complex_example.pxc
ProbWorks Manual\	Probworks_v1.1_manual.pdf
ProbWorks Licenses\	ProbWorks_SingleUser_License.pdf ProbWorks_Division_License.pdf

REGISTRATION

Pi Blue Software, Inc.. maintains a database of active users of ProbWorks. Registered users are notified of product updates, improvements, and bug fixes. To be added to the users database send an email with name, address, and phone number to the Technical Support Group at techsupport@piblue.com.

ADDITIONAL RESOURCES

Technical Support

Technical support for ProbWorks is available by email Monday through Friday 9:00 AM to 5:00 PM EST. Send a brief description of the problem in question as well as the particular software configuration (operating system version, ModelCenter© version, Analysis Server© version) to techsupport@piblue.com.

Using Online Support

Pi Blue Software maintains a FAQ support page at http://www.piblue.com/products/probworks. Common questions related to the installation of the individual ProbWorks tools are posted and maintained in the FAQ.

Consulting Referral Services

Pi Blue Software also offers engineering consulting services to help users maximize their productivity with ProbWorks. Typical services include on-site training, model and application support, and domain-specific problem setup and analysis. This consulting support is available on an hourly, daily, or fixed-price basis. Contact SEI at info@piblue.com to obtain more information on consulting referral services.

CONVENTIONS USED IN THIS MANUAL

This manual uses the following conventions:

- References to file paths appear in bold Times Roman font, such as: **c:\program files**
- References to variable names appear in bold Arial font, such as: **numberOfTrials**
- References to component blocks appear in italic Times Roman font, such as: *runVariables*
- References to file folder names appear in italic Arial font, such as: *Examples*

CHAPTER 1: GENERAL FUNCTIONALITY AND SETUP

BASIC FUNCTIONALITY

Each component in the ProbWorks suite is a type of procedure designed to assist with probabilistic uncertainty analysis. These components perform different analyses but share some basic functionality. Using similar variables, parameters, and interfaces; the user is able to develop familiarity and transition between components with ease (including the use of a multiple components on the same analysis). This section covers the general installation and operation of the ProbWorks component suite. Each component is a set of compiled Java programs in a file archive, allowing them to run on any platform that has Analysis Server© installed. An exception is the ProbWorks Graph Plug-in utility. This program is a ModelCenter© plug-in which provides a graphing capability for additional analysis of the outputs from other ProbWorks components.

COMPONENTS OF THE PROBWORKS SUITE

The algorithmic background and operational strength of each ProbWorks component are described in this section. Knowledge of these characteristics is useful for selecting a component for a particular uncertainty analysis problem. More details on each component can be found in their respective sections.

- **ProbWorks_MonteCarlo**
  This component performs Monte Carlo uncertainty analysis. The user specifies distributions for input variables for the analysis and output variables to monitor. The component then performs a user-specified number of trials. For each trial, a single value is chosen for each input variable from its distribution, the model is run, and then the resulting outputs are recorded. Output distributions are formed from the set of variables that are monitored from each trial.

- **ProbWorks_DPOMD**
  This component performs an approximation of a Monte Carlo analysis using the technique known as Discrete Probability Optimal Matching Distributions (DPOMD). Approximations for normal distributions are applied to the input variables, specific runs are performed, and output normal distributions are reconstructed from the results. This approximation technique typically runs much faster than a traditional Monte Carlo analysis. The current implementation is limited to using normal input distributions.

- **ProbWorks_RSEgenerator**

  This component creates an approximation to any ModelCenter© component or model in the environment and stores this information for later use in the form analytical Response Surface Equations (RSEs). In particular, this component uses quadratic approximation and is useful for approximating models with long run times. This is very beneficial when using any type of Monte Carlo analysis that normally will require many trials.

- **ProbWorks_ParetoAnalysis**

  The Pareto analysis component determines the impact of any ModelCenter© component or model's input parameters on selected output parameters. This allows the analyst to determine the most important inputs of a component or model. Such information can be used to narrow down the number of input parameters to be used to generate a Response Surface Equation (RSE) and/or to be used for a Monte Carlo or DPOMD analysis.

- **ProbWorks_GraphPlugIn**

  The graph plug-in allows visualization of output data from ModelCenter© components or models. This plug-in can be coupled with the other components in the ProbWorks suite in order to organize and visualize output data. This add-in has the capability to specifically create either histograms or Pareto charts. For example, histograms can be created from Monte Carlo data allow the analyst to see the resulting distributions on output variables and get a visual indication of the impact of uncertainty on their model. Pareto charts can give a graphical representation of the impact of each input variable on an output variable.

GENERAL COMPONENT OPERATION

The analysis components in the ProbWorks suite, separate from the plug-in components, all follow a similar method of operation that is outlined below:

1. **Place the component in the model to be analyzed:**
   a. Open an existing model in ModelCenter© or create a new one.
   b. Connect to the Analysis Server© with ProbWorks installed.
   c. Drag the ProbWorks component from the Server Browser into the Analysis View.

2. **Set up the ProbWorks component variables:**
   a. Open up the driverControls subsection under the primary ProbWorks component in the Component Tree.
   b. Specify values for each of the various input parameters. Holding the mouse pointer over an input name for a few seconds will bring up information about that input along with a description, applicable bounds, and recommended values.
3. **Set up the design variables and forecast or response variables:**
   a. Double click on the ProbWorks component to open the Generic Driver Tool.
   b. Drag inputs from the Component Tree into the *designVariables* block in the Generic Driver Tool and set up the parameters for each variable.
   c. Drag outputs from the Component Tree into the *forecastVariables* or *responseVariables* block in the Generic Driver Tool and set up the parameters for each variable.

Setup of the ProbWorks component is now complete. Pressing the Run button will then initiate the analysis, returning the results in its variable tree as well as the Data Collector if used.

The graph plug-in component is run from the Data Collector, and its usage is described in Chapter 6.

PROBWORKS INTERFACE

All of the ProbWorks components share a similar interface within ModelCenter©. There are four basic windows in the environment: the Server Browser, the Analysis View, the Component Tree, and the Generic Driver Tool. Figure 1 shows the ProbWorks_MonteCarlo component connected to a simple example and highlights all of these features.

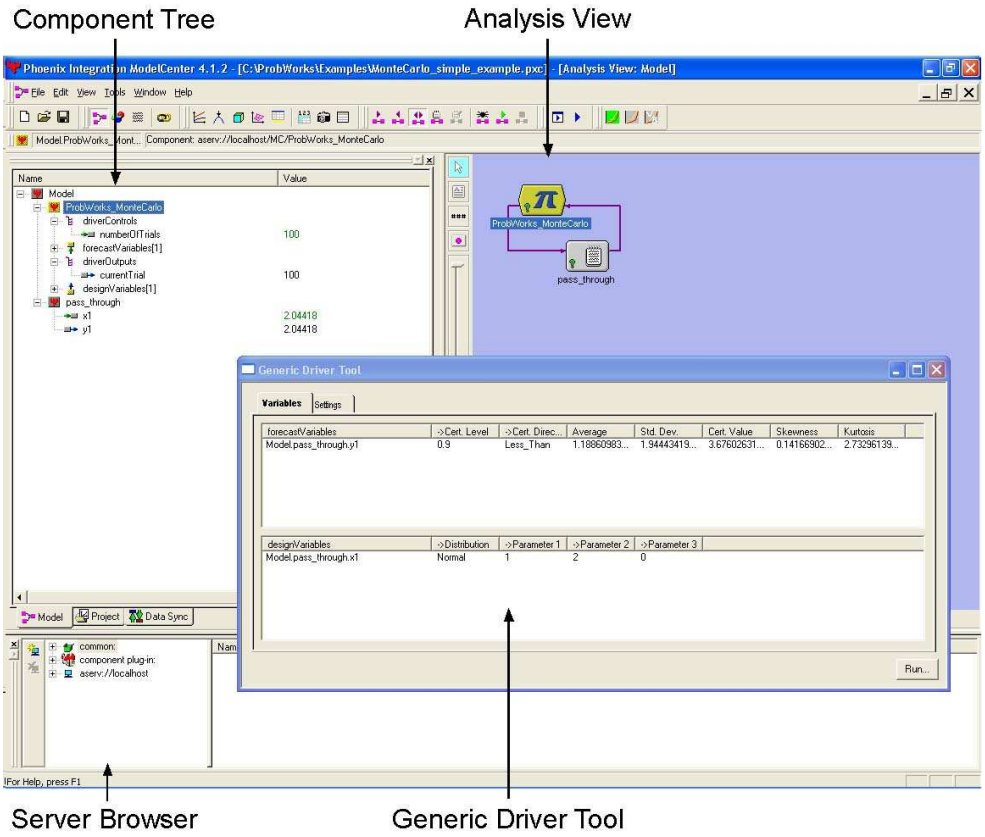


Figure 1: Monte Carlo Component Interface Elements

The Server Browser, located in the bottom of the ModelCenter© window, shows the available list of Analysis Server© machines. Clicking on the ProbWorks folder on the Analysis Server© with ProbWorks installed will bring up a list of the ProbWorks analysis components. Dragging one of these into the analysis view with the model to be analyzed creates an icon of the ProbWorks component.

The Analysis View, in the center of the ModelCenter© window, provides a graphical representation of the model. Once the design variables and response or forecast variables have been specified the Analysis View will show the analysis component linked to the model to be analyzed.

The Component Tree, located on the left of the ModelCenter© window, shows a list of all the current model's components and their variables. Underneath the analysis component is a list of all of the driver's controls and outputs. This section can be used to specify inputs and view outputs during and after the run.

Double clicking on the ProbWorks component in the Analysis View opens the Generic Driver Tool window. Design variables and response or forecast variables can be dragged from the component's variables in the Component Tree into their relevant section in the Generic Driver Tool. The inputs for each design variable and forecast or response variable can then be specified, each input for these variables is identified with an arrow, "->" in their title. Clicking on the "run" button in the Generic Driver Tool starts the analysis procedure. This also brings up the ModelCenter© Data Collector that stores all of the function calls made during the analysis and can be used for graphing the results or saving them to a text file.

Error messages are returned to ModelCenter© by ProbWorks components as IllegalArgumentExceptions and appear in Figure 2. These error messages provide information about what caused the error and possible solutions. The ProbWorks component will not run until the error is corrected. Each component's section provides a table with a list of all error messages for that component.

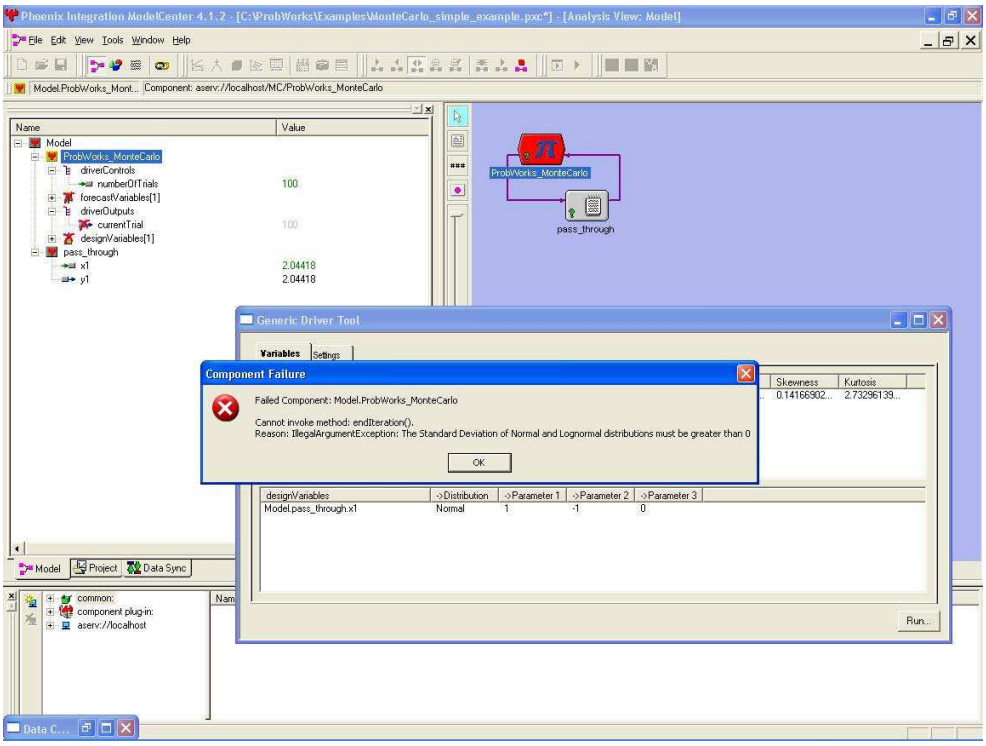


Figure 2: Monte Carlo Component Sample Error Message

PROBWORKS INSTALLATION AND SAMPLE USAGE

This section illustrates the setup and use of a ProbWorks component (in this case the Monte Carlo component). The example shown was performed on a PC running Microsoft

Windows XP with Phoenix Integration's Analysis Server© 3.12 and ModelCenter© 4.1.2. First ensure that the ProbWorks_MonteCarlo component is installed and that the Analysis Server© is subsequently initiated. All of the ProbWorks analysis components are installed by placing them in a directory accessible by Analysis Server© on any machine with Analysis Server© installed such as the analyses directory in the Analysis Server© installation directory. For example, on a Microsoft Windows system the default directory would be:

c:\program files\Phoenix Integration\AnalysisServer 3.1\analyses

It is in this directory that the ProbWorks folder should be placed. For this case the ProbWorks_MonteCarlo component is based upon a Java file called ProbWorks_MonteCarlo.jar. Subsequently, the ProbWorks_MonteCarlo component would then be located at:

c:\program files\Phoenix Integration\AnalysisServer 3.1
analyses\ProbWorks\ProbWorks_MonteCarlo.jar

The ProbWorks_GraphPlugIn component is installed through the ModelCenter© 4.0+ plug-in manager interface. To install this component, open up ModelCenter©, then open up the Plug-In Manager by selecting Tools in the menu, then select the Plug-In Manager. Click on the Data Collector Plug-Ins folder, then on the Java tab. Fill out the form as follows:

Java Plug-In Class Name: ProbWorks_GraphPlugIn
.jar File Path: the path to the ProbWorks_GraphPlugIn .jar file, such as
c:\program files\Phoenix Integration\AnalysisServer 3.1\analyses
ProbWorks\ProbWorks_GraphPlugIn.jar
Java Plug-In Class Name: ProbWorks_GraphPlugIn
Icon Location: the path to an icon file, such as
c:\program files\Phoenix Integration\AnalysisServer 3.1\analyses
ProbWorks\PBS_logo.bmp

Finally, click the Add Plug-In button to add the plug-in to ModelCenter©. The plug-in will now be available from the Data Collector, visually defined by the SpaceWorks logo.

Figure 3 shows the ModelCenter© 4.1 environment, as it will appear for this example problem once the ProbWorks_MonteCarlo component has been added, the variables set up, Generic Driver Tool opened, and the analysis performed.

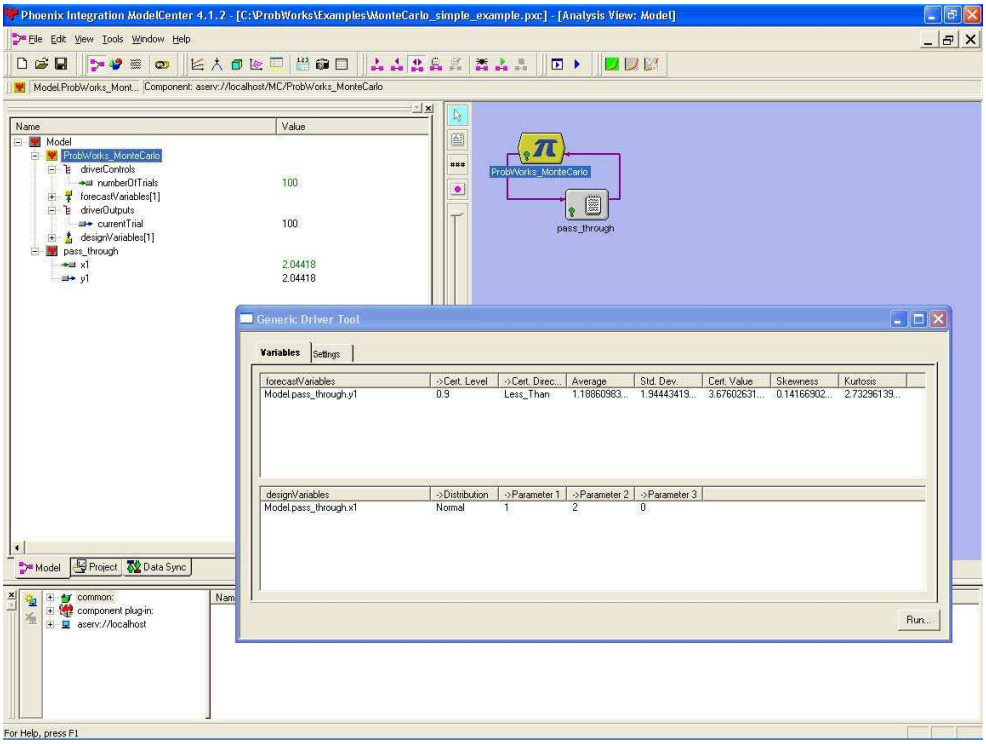


Figure 3: Sample Monte Carlo Setup

In order to see this example the user can open the simple_example.pxc in ModelCenter©. The file is located on the ProbWorks installation CD under the *ProbWorks Example Problems* directory, *Generic Example Scripts Used* subdirectory. This file contains a single script, pass_through, which specifies one input, x1 and a single output, y1, where y1 = x1. Figure 3 shows this file with the Monte Carlo component added.

Locate the local Analysis Server© on the Server Browser (bottom left pane in ModelCenter©). This server is usually called localhost. If localhost is not present, add it by clicking on the Add Server icon in the top right of the Server Browser window. Click on the ProbWorks directory to bring up a list of the ProbWorks analysis components to the right. Drag the ProbWorks_MonteCarlo component from the list into the model near the pass_through component.

Along the left side of ModelCenter©, the Component Tree displays the current components in the model. Expanding this tree provides access to each of the component's variables. For example, by expanding the pass_through component, its input and output are shown with their current values. Every ProbWorks analysis component has such driver controls and driver outputs, some components may also have additional subsections where needed.

A sample execution of one of the ProbWorks components is given below. This example is meant to illustrate the process used to implement a component. The specific example given uses the ProbWorks_MonteCarlo component that performs a Monte Carlo uncertainty analysis.

In a Monte Carlo simulation, inputs to a model are represented as distributions representing the various types of uncertainty in their values. The Monte Carlo component in ProbWorks performs a series of trials in which a random value is chosen for each input variable based on a distribution. The model is then run for a specified number of trials and the values for key forecast output parameters are recorded. The analysis then calculates statistics for each of the forecast variables. In this example the **numberOfTrials** variable could be set to 100. This is lower than the number of trials necessary for a credible Monte Carlo simulation for most models but does provide an accurate enough gauge of how the component operates.

The design variables and forecast variables are then specified. Double clicking on the Monte Carlo component brings up the Generic Driver Tool that has a *forecastVariables* block and a *designVariables* block. Drag the output, y1, from beneath the pass_through script on the Component Tree, to the *forecastVariables* block and set the **Cert. Level** (certainty level) to 0.9 and the **Cert. Direction** (certainty direction) to "Less_Than". In addition to statistics about the shape of the output distribution, ProbWorks_MonteCarlo will calculate the point at which some percentage of the outputs is greater than or less than a specified percentage of all output values. For this example, a certainty level of 0.9 will output a point for which 90% of the outputs are less than that value, indicating a 90% confidence that the certainty level can be achieved.

Next, drag the pass_through example input, x1, into the *designVariables* block and then set the **Distribution** to "Normal", **Parameter 1** to 1, **Parameter 2** to 2, and **Parameter 3** to 0. This will set the Monte Carlo to use a normal distribution for the input variable with a mean of 1 and a standard deviation of 2, the first two parameters being used to define the distribution in this case. Other distributions will use these three parameters in different ways. Further parts of this manual provide explanations on how to input other distributions using these parameter values.

At this point, the Monte Carlo sample problem is now ready to be executed and should look similar to Figure 3. Click on the green arrow on the ProbWorks_MonteCarlo component or click Run on the Generic Driver Tool. If the component was run from the

Generic Driver Tool, the Data Collector will open and record all of the variable values for each run of the model. The ProbWorks_MonteCarlo component will display the current trial it is performing in the *driverOutputs* section of its variable tree and the current output statistic for the **forecastVariable** in the Generic Driver Tool. Once the component has finished running (the components will stop flashing) the **Average** should be approximately 1 and the **Std. Dev.** (standard deviation) approximately 2, as shown in Figure 3. Note that the values will not be exact, as the Monte Carlo analysis is by its design probabilistic so each run will be slightly different from the last.

All of the ProbWorks analysis components are set up and run in a similar fashion. For details on each component refer to the appropriate section in this manual. The ProbWorks_GraphPlugIn component is run from the Data Collector, as described in Chapter 6.

CHAPTER 2: PROBWORKS – MONTE CARLO

OVERVIEW

The ProbWorks_MonteCarlo component performs a Monte Carlo analysis on a specified model. This component is useful for determining the variation of key metrics or variables of a model due to uncertainty in primary design variables or other key noise variables.

MATHEMATICAL FOUNDATION

Monte Carlo techniques are based on variety of numerical methods that use sequences of random numbers to perform a statistical simulation. For each trial the design variables are set to random values based on selected input distributions. The value of each forecast variable is recorded and certain key output statistics are calculated. For a large number of trials, the design variables will begin to approximate the selected distributions and the forecast variables will show the effect of the design variables' variation.

The ProbWorks_MonteCarlo component performs Monte Carlo analyses on selected forecast variables by varying given design variables. The user specifies both the design variables along with the associated distributions. The driver component collects data from forecast variables and maintains running calculations of mean, standard deviation, certainty values, skewness, and kurtosis for given certainly levels. Figure 4 shows an example frequency distribution produced in ModelCenter© by the Monte Carlo component for 1000 trials of a normal distribution with a mean of 1 and a standard deviation of 2.

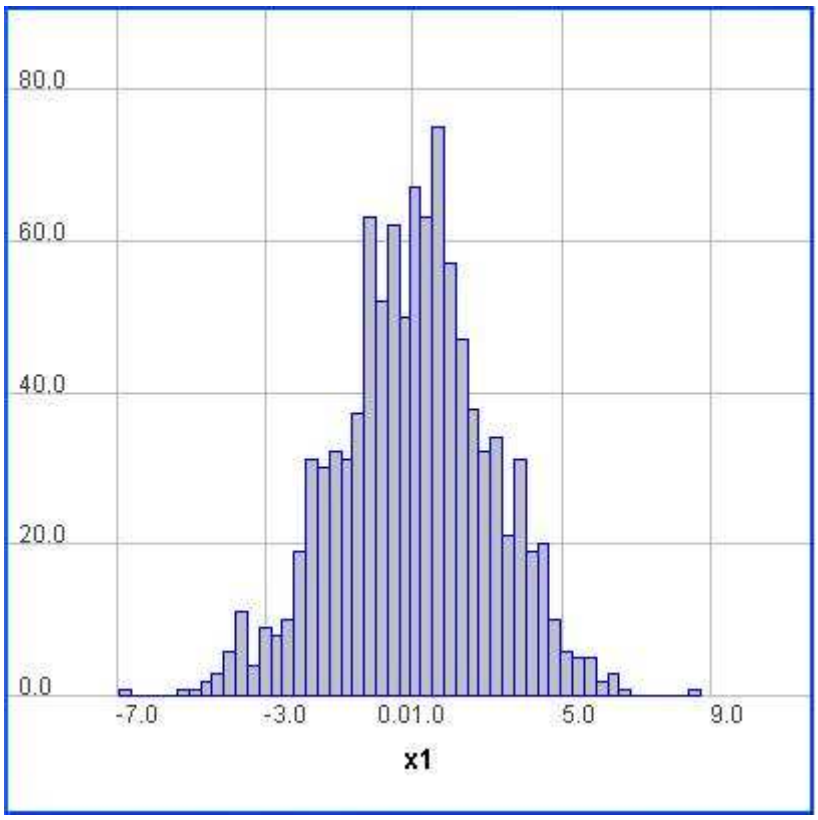


Figure 4: Example Monte Carlo Distribution

INPUT-OUTPUT VARIABLES

Inputs

Table 3 lists all of the inputs along with their type and available values.

Table 3. ProbWorks_MonteCarlo Inputs

VARIABLE OR SUBSECTION	TYPE	AVAILABLE VALUES
DriverControls		
numberOfTrials	Integer	≥ 0
forecastVariables	Double	Any
Cert. Level	Double	0 to 1
Cert. Direction	String	Greater_Than, Less_Than
designVariables	Double	Any
Distribution	String	Normal, Uniform, Triangle, Exponential, Weibull, Lognormal, Beta
Parameter 1	Double	Distribution Dependant
Parameter 2	Double	Distribution Dependant
Parameter 3	Double	Distribution Dependant

Driver Controls

These inputs are specified in the *driverControls* subsection of the ProbWorks_MonteCarlo variable tree in the ModelCenter© Component Tree.

- **numberOfTrials**: the number of trials to perform for the Monte Carlo analysis. Values for this variable are problem dependant: enough trials are necessary to statistically capture the response distribution; nominally 1,000 to 10,000 trials are good measures in terms of the number of trials.

Forecast Variables

Forecast variables are specified through dragging outputs from the model in the Component Tree into the *forecastVariables* block of the Generic Driver Tool. Double-clicking on the component accesses this tool window. Each forecast variable also has inputs to be specified.

- **Cert. Level**: Each forecast variable has a desired certainty level input between 0 and 1. This value is used to calculate the certainty value, a proxy for the percentage of points less than or greater than this value. The certainty level is a decimal representing the percent of the data found on one side of the certainty value boundary. A typical value for the certainty level would be 0.9 to determine the value at which 90% of the points are greater than or less than the returned certainty value.

- **Cert. Direction**: Each forecast value has a certainty direction that specifies to which side of the certainty value data lies. If Greater_Than is selected with a certainty level of 0.9, then 90% of the data will be greater than the returned certainty value.

Design Variables

Design variables are specified through dragging inputs from the model in the Component Tree into the *designVariables* block of the Generic Driver Tool. Double-clicking on the Monte Carlo component accesses this tool window. Each design variable also has inputs to be specified.

- **Distribution:** Each design variable is assigned a distribution type. The available choices and their associated probability distribution functions are[1]:

  - **Normal:** $f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-(x-\mu)^2/2\sigma^2}$

    where σ is the standard deviation and μ is the mean

  - **Uniform:** $f(x) = \frac{1}{Max - Min}$ if Min < x < Max

    OR $f(x) = 0$ otherwise

  - **Triangle:** $f(x) = \frac{h(x - Min)}{Likeliest - Min}$ if Min < x < Likeliest

    OR $f(x) = \frac{h(x - Likeliest)}{Likeliest - Max}$ if Likeliest < x < Max

    OR $f(x) = 0$ otherwise

    where $h = \frac{2}{Max - Min}$

  - **Exponential:** $f(x) = \lambda e^{-\lambda x}$ if x $\geq$ 0 or $f(x) = 0$ if x < 0

    where λ is the rate

  - **Weibull:** $f(x) = \alpha\beta x^{\beta-1} e^{-\alpha x^{\beta}}$

    where α is the scale and β is the shape

- **Lognormal:** $f(x) = \frac{1}{\sigma x\sqrt{2\pi}} e^{(\ln(x)-\mu)^2/2\sigma^2}$

  where is the standard deviation and is the mean

- **Beta:** $f(x) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} x^{\alpha-1}(1-x)^{\beta-1}$

  where is the alpha parameter and is the beta parameter

- **Parameter 1:** Each design variable's distribution requires a certain number of parameters. Each parameter represents a different variable for different distributions all of the distributions and what each parameter represents are listed in Table 4. Parameter one represents:
  - mean for a normal distribution,
  - minimum value for a uniform distribution,
  - minimum value for a triangle distribution,
  - rate for an exponential distribution,
  - scale for a Weibull distribution,
  - mean for a lognormal distribution,
  - alpha for a Beta distribution.
- **Parameter 2:** Parameter two represents:
  - standard deviation for a normal distribution,
  - maximum value for a uniform distribution,
  - likeliest value for a triangle distribution,
  - shape for a Weibull distribution,
  - not used for an exponential distribution,
  - standard deviation for a lognormal distribution,
  - beta for a Beta distribution.
- **Parameter 3:** Parameter three represents:
  - maximum value for a triangle distribution,
  - not used for any other distributions.

Table 4. Parameters to Define Various Distributions

DISTRIBUTION	PARAMETER 1	PARAMETER 2	PARAMETER 3
Normal	Mean	Standard deviation	Not used
Uniform	Minimum value	Maximum value	Not used
Triangle	Minimum value	Likeliest value	Maximum value
Exponential	Rate	Not used	Not used
Weibull	Scale	Shape	Not used

Outputs

Driver Outputs

These outputs are displayed in the *driverOutputs* subsection of the component's variable tree in the ModelCenter© Component Tree.

- **currentTrial**: the current try the Monte Carlo analysis is performing.

Forecast Variables

These outputs are displayed in the Generic Driver Tool, which is accessed by double clicking on the Monte Carlo component. The component calculates output statistics for each forecast variable while it is running. The statistics displayed are[2]:

- **Average:** The average value for each forecastVariable over the series of trials.
- **Std. Dev.:** The standard deviation for each forecastVariable.
- **Cert. Value:** Each forecast value has a certainty direction that specifies to which side of the certainty value data lies. If Greater_Than is selected with a certainty level of 0.9, then 90% of the data will be greater than the returned certainty value.
- **Skewness:** An estimate of the Fisher skewness for the variable. A negative skewness indicates the left tail of the distribution is longer while a positive skewness indicates the right tail is longer
- **Kurtosis:** An estimate of the Pearson kurtosis. Kurtosis represents the "peakedness" of a distribution; a kurtosis greater than three indicates a high peak while less than three indicates a flat top.

Error Messages

Some input variable settings will cause error messages. Table 5 lists all of the error messages and what should be done to resolve the problem.

Table 5. ProbWorks_MonteCarlo Error Messages and Resolutions

ERROR MESSAGE	RESOLUTION
At least one designVariable must be specified: double click the ProbWorks_MonteCarlo component to open the Generic Driver Tool then drag input variables from other components into the designVariables box and set up the distribution and its parameters	Add a design variable
At least one forecastVariable must be specified: double click the ProbWorks_MonteCarlo component to open the Generic Driver Tool then drag output variables from other components into the forecastVariables box and set up the certaintyLevel and its direction	Add a forecast variable
numberOfTrials must be greater than 0	Set the numberOfTrials variable to a value greater than 0
The Standard Deviation of Normal and Lognormal distributions must be greater than 0	Set the Parameter 2 variable for any Normal or Lognormal distribution to a value greater than 0
The High Bound must be greater than or equal to the Most Likely Value which must be greater than or equal to the Low Bound for a Triangle distribution	Set the Parameter 3 variable for any Triangle distributions to a value greater than its Parameter 2 value, and its Parameter 2 variable to a value greater than its Parameter 1 value
The High Bound must be greater than the Low Bound for Triangle and Uniform distributions	Set the Parameter 3 variable for any Triangle distributions to a value greater than its Parameter 1 value and the Parameter 2 variable for any Uniform distributions to a value greater than its Parameter 1 value
Alpha, Beta, Theta, and Rate for Weibull, Beta, and Exponential distributions must be greater than zero	Set the appropriate Parameter to a value greater than 0 for any Weibull, Beta, or Exponential distributions

SAMPLE USAGE

Simple Analysis: Analytical Function

This example demonstrates connecting the ProbWorks_MonteCarlo component to a simple model. The example is located in the ModelCenter© file MonteCarlo_simple_example.pxc located in the *ProbWorks Example Problems* directory, *Monte Carlo Examples* subdirectory on the ProbWorks CD. This file contains the ProbWorks_MonteCarlo component connected to a script, pass_through, which has one input, x1, and a single

output, y1, where y1 = x1. Figure 5 shows this example with inputs and outputs displayed from a run of the analysis. Table 6 gives the values for all input variables and Table 7 lists the values for all output variables after the analysis has been run.

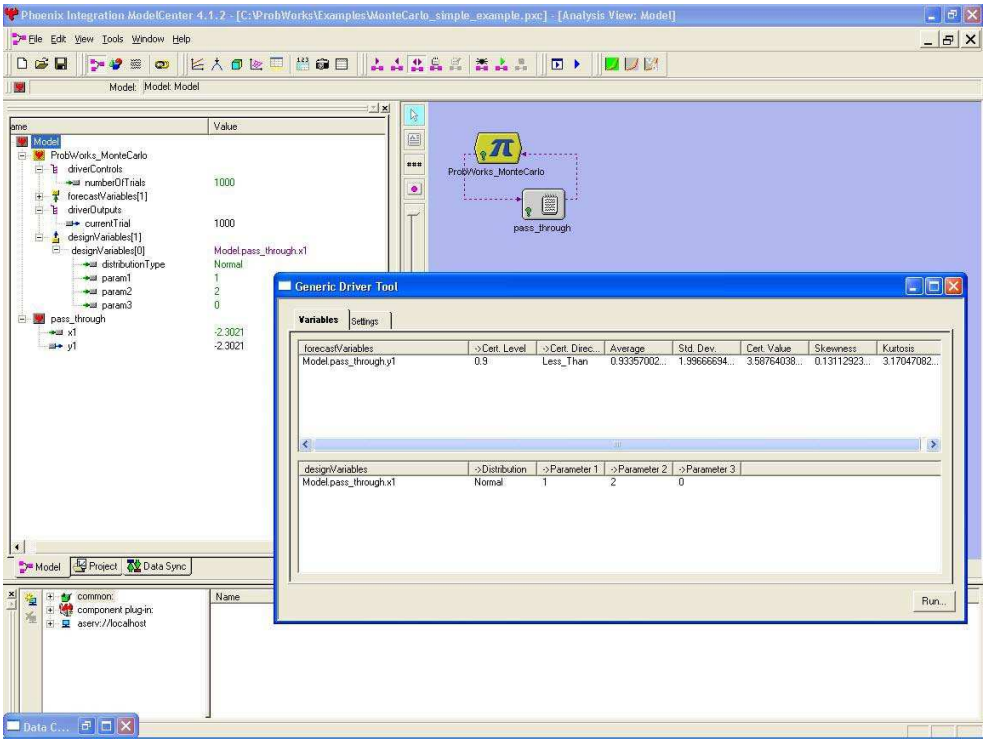


Figure 5: ProbWorks_MonteCarlo Simple Example

Table 6. ProbWorks_MonteCarlo Simple Example Inputs

VARIABLE OR SUBSECTION	VALUE
DriverControls	
numberOfTrials	1000
forecastVariables	
forecastVariables[0]	Model.pass_through.y1
Cert. Level	0.9
Cert. Direction	Less_Than
designVariables	
designVariables[0]	Model.pass_through.x1
Distribution	Normal
Parameter 1	1
Parameter 2	2
Parameter 3	0

Table 7. ProbWorks_MonteCarlo Simple Example Outputs

VARIABLE OR SUBSECTION	VALUE
driverOutputs	
currentTrial	1000
forecastVariables	
forecastVariables[0]	
Average	0.9335…
Std. Dev.	1.9966…
Cert. Value	3.5876…
Skewness	0.1311…
Kurtosis	3.1704…

The output distribution is exactly the same as the input distribution, giving an example of the characteristics of a distribution produced by the Monte Carlo component. Due to the probabilistic nature of a Monte Carlo analysis, each run will be slightly different, but with enough trials, a close approximation to a normal distribution will emerge. As the results show, the average for the results was close to that specified of 1, and the standard deviation

was close to the specified 2. In addition, the skewness was close to 0 and the kurtosis close to 3, as expected for a normal distribution.

Complex Analysis: Analytical Function

This example demonstrates connecting the ProbWorks_MonteCarlo component to a more complex function. The example is located in the ModelCenter© file MonteCarlo_complex_example.pxc located in *the ProbWorks Example Problems* directory, *MonteCarlo Examples* subdirectory on the ProbWorks CD. This file contains the ProbWorks_MonteCarlo component connected to a script, combine_inputs, which has three inputs, x1, x2, and x3 and two outputs, y1 and y2, where y1=x1+x2+x3 and y2=x1*x2*x3. Multiple types of input distributions are used and the resulting output distributions are a combination of these inputs. Figure 6 shows this example with inputs and outputs displayed from a run of the analysis. Table 8 gives the values for all input variables and Table 9 lists the values for all output variables after the analysis has been run.

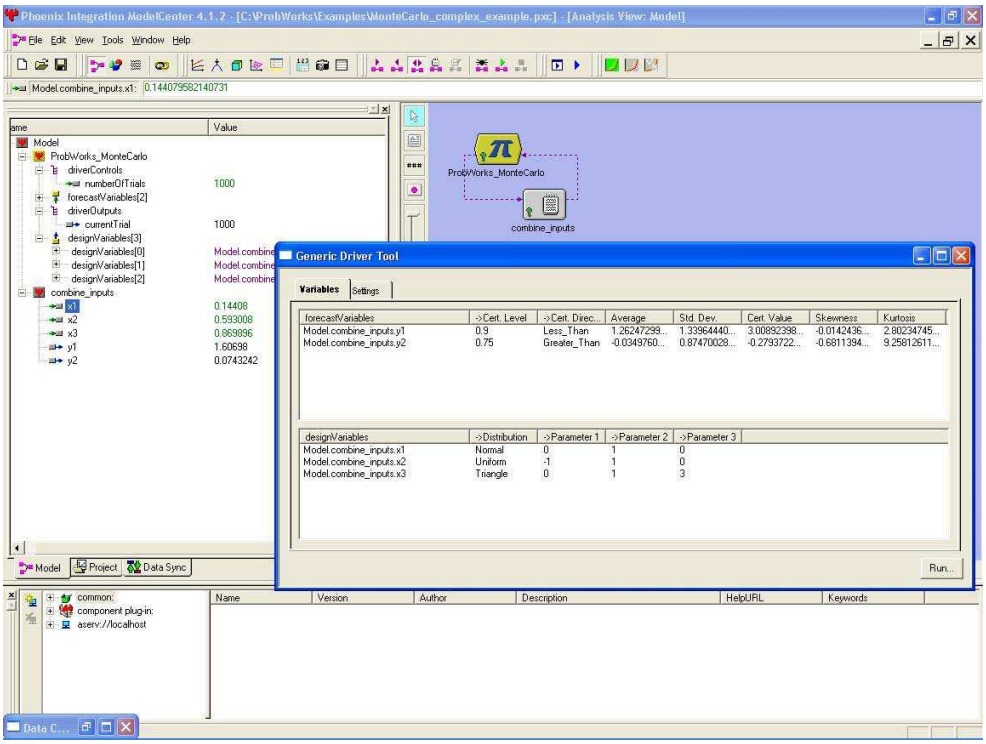


Figure 6: ProbWorks_MonteCarlo Complex Example

Table 8. ProbWorks_MonteCarlo Complex Example Inputs

VARIABLE OR SUBSECTION	VALUE
driverControls	
numberOfTrials	1000
forecastVariables	
forecastVariables[0]	Model.combine_inputs.y1
Cert. Level	0.9
Cert. Direction	Less_Than
forecastVariables[1]	Model.combine_inputs.y2
Cert. Level	0.75
Cert. Direction	Greater_Than
designVariables	
designVariables[0]	Model.combine_inputs.x1
Distribution	Normal
Parameter 1	0
Parameter 2	1
Parameter 3	0
designVariables[1]	Model. combine_inputs.x2
Distribution	Uniform
Parameter 1	-1
Parameter 2	1
Parameter 3	0
designVariables[2]	Model. combine_inputs.x3
Distribution	Triangle
Parameter 1	0
Parameter 2	1
Parameter 3	3

Table 9. ProbWorks_MonteCarlo Complex Example Outputs

VARIABLE OR SUBSECTION	VALUE
driverOutputs	
currentTrial	1000
forecastVariables	
forecastVariables[0]	
Average	1.2624…
Std. Dev.	1.3396…
Cert. Value	3.0089…
Skewness	-0.0142…
Kurtosis	2.8023…
forecastVariables[1]	
Average	-0.0349…
Std. Dev.	0.8747…
Cert. Value	-0.2793…
Skewness	-0.6811…
Kurtosis	9.2581…

The results show that the distribution created by adding together inputs had a wider range than any of the individual inputs and a higher standard deviation than the second distribution. The second distribution had a very narrow peak, with a high kurtosis and low standard deviation.

CHAPTER 3: PROBWORKS – DPOMD

OVERVIEW

The ProbWorks_DPOMD component encapsulates the Discrete Probability Optimal Matching Distributions (DPOMD) technique, providing an approximation to a Monte Carlo analysis. The component applies normal distributions to specified input variables and uses particular combinations of those variables to produce outputs from which statistics can be generated. This method typically requires many fewer runs than a Monte Carlo analysis, and is therefore beneficial when executing models that require long recalculation times.

MATHEMATICAL FOUNDATION

Monte Carlo methods are best implemented where the computational expense of performing numerous analyses is low. As the number of random runs increases, the accuracy of the Monte Carlo results increase. However, for difficult or time consuming engineering analysis problems, approximate methods are often used to reduce the computational expense of performing uncertainty analysis. Approximate methods such as response surface methods can be used to fit a very fast acting surrogate model of the true analysis. Alternately, methods that retain the full analysis, but use limited samples from the input distributions to approximate the output statistics can be used. Kriging methods, Design of Experiments (DoE) methods, and Descriptive Sampling are candidate methods from this class. Discrete Probability Optimal Matching Distribution (DPOMD) is a new method that compares favorably to these approaches with respect to both computing time and accuracy[3].

The effectiveness of DPOMD is fundamentally enabled by the process of reducing a continuous multivariate input distribution to a discrete distribution, as shown in Figure 7. A two-level fractional factorial design of experiments is used to produce a discrete distribution with zero mean vector and identity covariance matrix in standardized normal space[4]. Using values for the mean and covariance matrix from the input distribution, an inverse Hasofer-Lind transform is utilized to map the discrete distribution to the problem space[4]. In this way the simplified discrete distribution in the problem space maintains the same first moment (mean) and second moment (standard deviation) properties as the continuous input. A simulation may then be executed at these discrete points to produce an output distribution.

The ProbWorks_DPOMD component performs a fractional factorial DPOMD analysis on selected forecast variables by varying supplied design variables with approximated normal

distributions. The current implementation assumes that both the input and output distributions are symmetric and follow a normal distribution. This approximation gets better as the number of inputs increases. During problem setup, input or 'design' variables are specified, each with a corresponding mean and standard deviation. Once the required analyses are performed, the data from the forecast variables is used to calculate the mean, standard deviation, and certainty values for given certainty levels (e.g. the 90th percentile result). ProbWorks_DPOMD allows the user to specify the use of full factorial or fractional factorial experimental arrays to determine the required discrete input values. The fractional factorial array option allows more design variables to be specified without an increase in the number of runs.

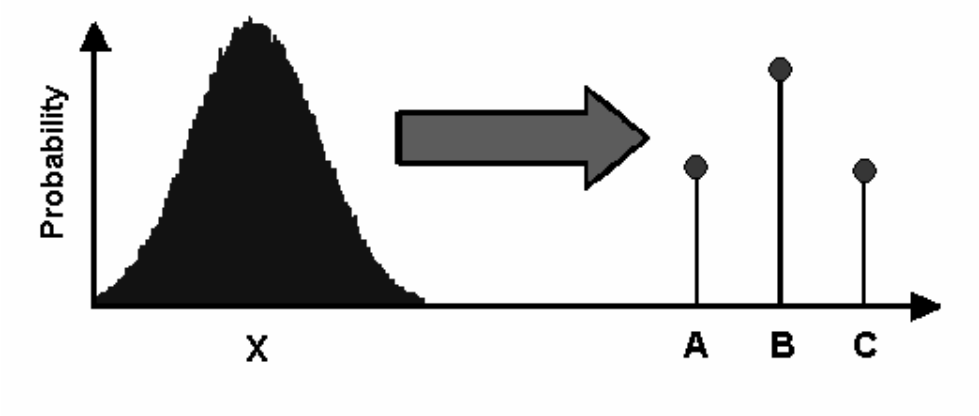


Figure 7: Illustration of Discrete Distribution Derived from Continuous Distribution[4]

INPUT-OUTPUT VARIABLES

Inputs

Table 10 lists all of the inputs along with their type and available values.

Table 10. ProbWorks_DPOMD Inputs

VARIABLE OR SUBSECTION	TYPE	AVAILABLE VALUES
driverControls		
covariance	Double array	Either blank or comma, space, or semicolon delimited array of doubles
reduction	Integer	≥ 0
forecastVariables	Double	Any
Cert. Level	Double	0 to 1
Cert. Direction	String	Less_Than, Greather_Than
designVariables	Double	Any
Mean	Double	Any
Std. Dev.	Double	> 0

Driver Controls

These inputs are specified in the *driverControls* subsection of the ProbWorks_DPOMD variable tree in the ModelCenter© Component Tree.

- **covariance**: the covariance array for the design variables. Normally this will be left blank and the component will create a default covariance array setting each input variable's variance to 1 and the covariance between each input variable to 0. A comma, space, or semicolon-separated list of double-type variables can be used to specify the array. For example, with 2 design variables the default covariance array would be: 1,0;0,1 representing the array:

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

and for 3 variables: 1,0,0;0,1,0;0,0,1 representing the array:

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Usually this variable will be left blank causing the default to be used.

- **reduction**: the value by which the full factorial array will be reduced to a fractional factorial array. Each increment in the reduction value results in halving the number of cases to be run. Typically this value is left at 0, except

in cases where there are large numbers of design variables and increased run time is desired.

Forecast Variables

Objective functions are specified by dragging outputs from the model in the Component Tree into the *forecastVariables* block of the Generic Driver Tool. Double-clicking on the component accesses this tool window. Each objective function also has inputs to be specified.

- **Cert. Level**: Each forecast variable has a desired certainty level input between 0 and 1. This value is used to calculate the certainty value, a proxy for the percentage of points less than or greater than this value. The certainty level is a decimal representing the percent of the data found on one side of the certainty value boundary. A typical value for the certainty level would be 0.9 to determine the value at which 90% of the points are greater than or less than the returned certainty value.
- **Cert. Direction**: Each forecast value has a certainty direction that specifies to which side of the certainty value data lies. If Greater_Than is selected with a certainty level of 0.9, then 90% of the data will be greater than the returned certainty value.

Design Variables

Design variables are specified by dragging inputs from the model in the Component Tree into the *designVariables* block of the Generic Driver Tool. Double-clicking on the component accesses this tool window. Each design variable also has inputs to be specified.

- **Mean**: the mean value for this design variable's normal distribution.
- **Std. Dev.**: the standard deviation for this design variable's normal distribution.

Outputs

Driver Outputs

These outputs are displayed in the driverOutputs subsection of the component's variable tree in the ModelCenter© Component Tree.

- **currentRun**: the current run the DPOMD analysis is performing.
- **totalRuns**: the total number of runs to be performed.

Forecast Variables

These outputs are displayed in the Generic Driver Tool, which is accessed by double clicking on the DPOMD component. The component calculates output statistics for each forecast variable when it is finished running. The statistics displayed are:

- **Mean:** The mean value for each forecastVariable over the series of runs.
- **Std. Dev.:** The standard deviation for each forecastVariable.
- **Cert. Value:** Each forecast value has a certainty direction that specifies to which side of the certainty value data lies. If Greater_Than is selected with a certainty level of 0.9, then 90% of the data will be greater than the returned certainty value.

Error Messages

Some input variable settings will cause error messages. Table 11 lists all of the error messages and what should be done to resolve the problem.

Table 11. ProbWorks_DPOMD Error Messages and Resolutions

ERROR MESSAGE	RESOLUTION
At least one designVariable must be specified: double click the ProbWorks_DPOMD component to open the Generic Driver Tool then drag input variables from other components into the designVariables box and set the mean and standard deviation	Add a design variable
At least one forecastVariable must be specified: double click the ProbWorks_DPOMD component to open the Generic Driver Tool then drag output variables from other components into the forecastVariables box and set up the certaintyLevel and its direction	Add a forecast variable
The Standard Deviation must be greater than zero for all designVariables	Set all Std. Dev. Variables to a value greater than 0
Certainty Levels should be between 0 and 1	Specify each Cert. Level as a value between 0 and 1
The Covariance string should be a n by n array where n is the number of designVariables	Specify either a blank covariance value, or an array with the proper number of values (designVariables squared)

SAMPLE USAGE

Simple Analysis: Analytical Function

This example demonstrates connecting the ProbWorks_DPOMD component to a simple pass through model. The example is located in the ModelCenter© file

DPOMD_simple_example.pxc located in the *ProbWorks Example Problems* directory, *DPOMD Examples* subdirectory on the OptWorks CD. This file contains the ProbWorks_DPOMD component connected to a script, pass_through, which specifies one inputs, x1 and a single output, y1, where y 1= x1. Figure 8 shows this example with inputs and outputs displayed from a run of the analysis. Table 12 gives the values for all input variables and Table 13 lists the values for all output variables after the analysis has been run.

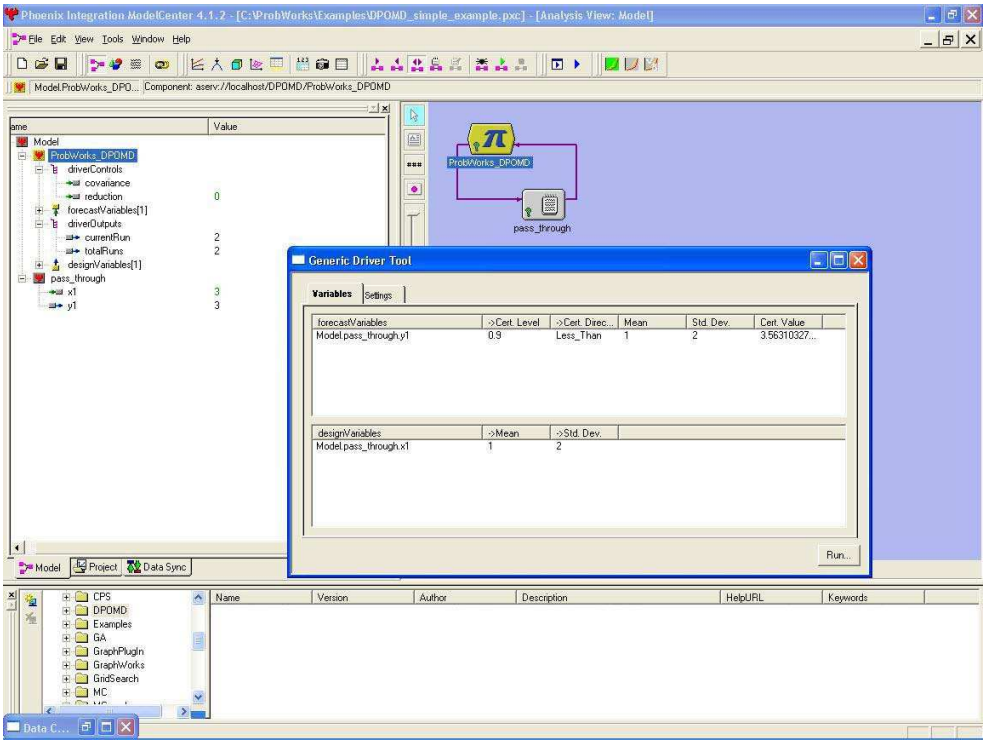


Figure 8: ProbWorks_DPOMD Simple Example

Table 12. ProbWorks_DPOMD Simple Example Inputs

VARIABLE OR SUBSECTION	VALUE
driverControls	
Covariance	
Reduction	0
forecastVariables	
forecastVariables[0]	Model.pass_through.y1
Cert. Level	0.9
Cert. Direction	Less_Than
designVariables	
designVariables[0]	Model.pass_through.x1
Mean	1
Std. Dev.	2

Table 13. ProbWorks_DPOMD Simple Example Outputs

VARIABLE OR SUBSECTION	VALUE
driverOutputs	
currentRun	2
totalRuns	2
forecastVariables	
forecastVariables[0]	
Mean	1
Std. Dev.	2
Cert. Value	3.5631…

For this simple problem, the DPOMD driver found an output distribution with exactly the same statistics as the input distribution, and ran much faster than the equivalent Monte Carlo run. For models with long run times in which normal distributions are adequate to represent the design variables' uncertainty, DPOMD can provide similar results to Monte Carlo analyses in significantly less time.

Complex Analysis: Analytical Function

This example demonstrates connecting the ProbWorks_DPOMD component to a more complex function. The example is located in the ModelCenter© file DPOMD_complex_example.pxc located in the *ProbWorks Example Problems* directory, *DPOMD Examples* subdirectory on the ProbWorks CD. This file contains the ProbWorks_DPOMD component connected to a script, combine_inputs, which has three inputs, x1,x2, and x3 and two outputs, y1 and y2 where y1=x1+x2+x3 and y2=x1*x2*x3. Figure 9 shows this example with inputs and outputs displayed from a sample run. Table 14 lists input variable values and Table 15 lists the output variable values for all after the analysis has been run.

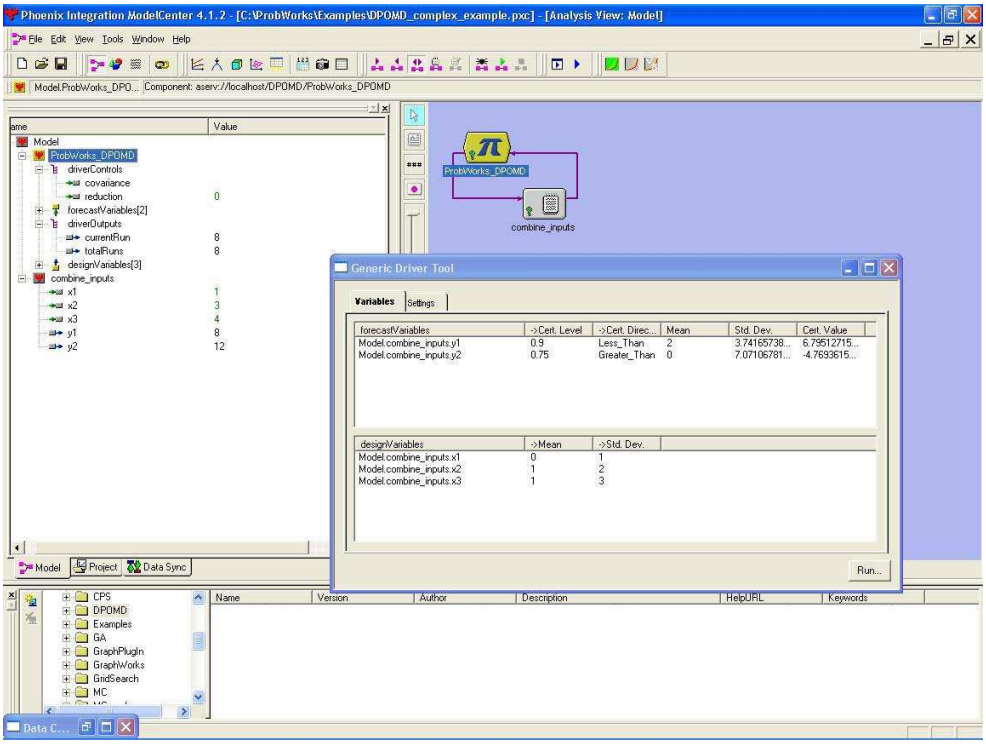


Figure 9: ProbWorks_DPOMD Complex Example

Table 14. ProbWorks_DPOMD Complex Example Inputs

VARIABLE OR SUBSECTION	VALUE
driverControls	
Covariance	
Reduction	0
forecastVariables	
forecastVariables[0]	Model.combine_inputs.y1
Cert. Level	0.9
Cert. Direction	Less_Than
forecastVariables[1]	Model.combine_inputs.y2
Cert. Level	0.75
Cert. Direction	Greater_Than
designVariables	
designVariables[0]	Model.combine_inputs.x1
Mean	0
Std. Dev.	1
designVariables[1]	Model.combine_inputs.x2
Mean	1
Std. Dev.	2
designVariables[2]	Model.combine_inputs.x3
Mean	1
Std. Dev.	3

Table 15. ProbWorks_DPOMD Complex Example Outputs

VARIABLE OR SUBSECTION	VALUE
driverOutputs	
currentRun	8
totalRuns	8
forecastVariables	
forecastVariables[0]	
Mean	2
Std. Dev.	3.7416…
Cert. Value	6.7951…
forecastVariables[1]	
Mean	0
Std. Dev.	7.0710…
Cert. Value	-4.7693…

The DPOMD analysis produced output statistics for the two distributions with fewer runs than a comparable Monte Carlo analysis. Since the Monte Carlo example shown in the previous chapter used distributions other than normal it could not be used in this case. The current implementation of the DPOMD driver only allows normal distributions.

CHAPTER 4: PROBWORKS – RSE GENERATOR

OVERVIEW

The ProbWorks_RSEgenerator component creates a Response Surface Equation (RSE) approximation to a model. The quadratic approximation is useful for many types of functions within limited bounds, providing a method to substitute slower executing computational codes with analytical approximations. The resulting RSE is placed in a ModelCenter©-runnable script that can be used in the current model file or placed in another.

MATHEMATICAL FOUNDATION

A Response Surface Equation (RSE) is an analytical equation that approximates a function. The ProbWorks_RSEgenerator component performs a specific set of runs based upon Design of Experiments (DOE) methodologies, in which each design variable is set to specific value and the values of all the response variables are recorded. A least squares fit is then used to create a quadratic equation that best represents the data collected. This equation is of the form:

$$\sum_{i=1}^{n} a_i x_i^{\,2} + \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} b_{ij} x_i x_j + \sum_{i=1}^{n} c_i$$

where n is the number of design variables, x_i and x_j are the design variables, a_i and a_j are the quadratic coefficients, b_{ij} are the cross term coefficients, and c_i are the constant coefficients. Figure 10 gives an example of this process.

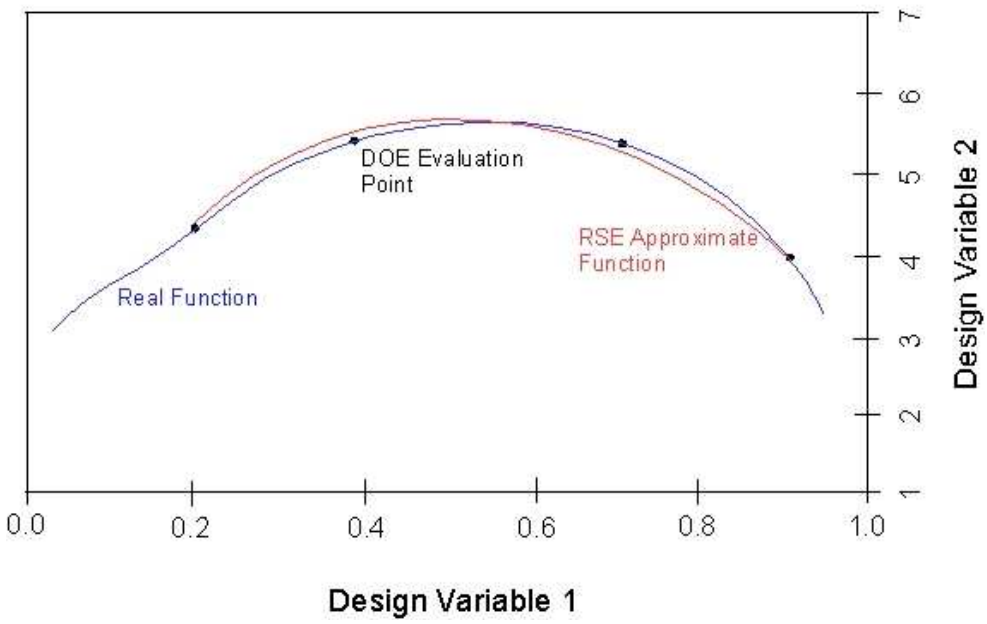


Figure 10: Example Response Surface Equation

INPUT-OUTPUT VARIABLES

Inputs

Table 16 lists all of the inputs along with their type and available values.

Table 16. ProbWorks_RSEgenerator Inputs

VARIABLE OR SUBSECTION	TYPE	AVAILABLE VALUES
DriverControls		
CCDstarPoint	Double	0 or ≥ 1
DOEtype	String	2nd Order CCD, 2nd Order D-Optimal, or Linear + X-Terms FF
RSEfileName	String	Any
responseVariables	Double	Any
designVariables	Double	Any
Low Setting	Double	Less than High Setting
High Setting	Double	Greater than Low Setting

Driver Controls

These inputs are specified in the driverControls subsection of the component's variable tree in the ModelCenter© Component Tree.

- **CCDstarPoint:** the type of DOE to use to generate the RSE: A Central Composite Design (CCD) which fits all second order terms, a D-Optimal design which fits all second order (max 15 design variables), or a Fractional Factorial (FF) design which fits all linear and mixed cross terms (but not pure quadratic terms).
- **DOEtype**: the star points' locations for a Central Composite DOE: this value is multiplied by the length of the fractional factorial cube's side to generate the extra star points for a Central Composite Design. When set to its default of 0, the value will be automatically generated for a rotatable array. If set to any other value, that value will be used, but must be equal to or greater than one.
- **RSEfileName**: the file name for the scriptWrapper component containing the RSE generated. If no name is specified the default 'RSEfile' will be used.

Response Variables

Response variables are specified by dragging outputs from the model in the Component Tree into the *responseVariables* block of the Generic Driver Tool. Double-clicking on the component accesses this tool window.

Design Variables

Design variables are specified by dragging inputs from the model in the Component Tree into the *designVariables* block of the Generic Driver Tool. Double-clicking on the component accesses this tool window. Each design variable also has inputs to be specified.

- **Low Setting**: the lowest setting in this variable's range to be used to generate the RSE.
- **High Setting**: the highest setting in this variable's range to be used to generate the RSE.

Outputs

Driver Outputs

These outputs are displayed in the driverOutputs subsection of the component's variable tree in the ModelCenter© Component Tree.

- **currentRun**: the current run the DPOMD analysis is performing.
- **numberOfCoefficients**: this is the total number of coefficients in the RSE.

- **totalRuns**: the total number of runs to be performed.
- **coefficients**: a list of the value of the coefficients, in the order: constant coefficients, cross term coefficients, then pure quadratic coefficients.
- **RSEfile**: a listing of the RSE file created which contains the RSE.

Response Variables

These outputs are displayed in the Generic Driver Tool, which is accessed by double clicking on the RSE generator component. The component calculates output statistics for each response variable when it is finished running. The statistics displayed are:

- **R squared:** the R^2 value for the fit of this variable. An R^2 value of 1 indicates a perfect fit between the RSE and the sample points taken from the actual function, the less an R^2 value is, the poorer the RSE's fit.
- **R Sq. Adj.:** the adjusted R^2 value for this variable, representing a good RSE fit with a value close to 1.

RSEfile

When the RSE generator component is run, it will create another ModelCenter© component with the default name of RSEfile (though other names can be specified with the RSEfile variable). This runnable component contains the RSE generated and can be found in the same folder as the ProbWorks_RSEgenerator that created it. To add this to a model, simply drag it into the model like a normal component. If updates are made to an RSEfile that already exists in a model, right click on the RSEfile component and select "Reconnect" to update its values.

The RSEfile has one input for each design variable, labeled x1, x2, … xn; and one output for each response variable, labeled y1, y2, … yn. It has an additional Boolean input, **allowExtrapolation** which is set to false by default. When set to false, the RSEfile will not allow design variable values with ranges outside those by which the RSE was originally created.

To use the RSEfile, simply connect it to other components in a model as if it were the original function it is replacing and it will run as a normal component.

Error Messages

Some input variable settings will cause error messages. Table 17 lists all of the error messages and what should be done to resolve the problem.

Table 17. ProbWorks_RSEgenerator Error Messages and Resolutions

ERROR MESSAGE	RESOLUTION
At least one designVariable must be specified: double click the ProbWorks_RSEgenerator component to open the Generic Driver Tool then drag input variables from other components into the designVariables box	Add a design variable
At least one responseVariable must be specified: double click the ProbWorks_RSEgenerator component to open the Generic Driver Tool then drag output variables from other components into the responseVariables box	Add a response variable
The responses have created a singular matrix, try changing the DOE type or design variables' ranges	Try changing the DOE type or the design variables' ranges
For a CCD design, the CCDstarPoint should either be 0 (to automatically generate the star points) or greater than or equal to 1	Set the CCDstarPoint variable to a value of either 0 or greater than or equal to 1 or change the DOEtype
All designVariable's high settings must be greater than their low settings	Set each design variable's High Setting to a greater value than its Low Setting
A D-Optimal design can have no more than 15 variables, remove some design variables or select a different type of DOE	Either remove some design variables or select a different DOEtype

SAMPLE USAGE

Simple Analysis: Analytical Function

This example demonstrates connecting the ProbWorks_RSEgenerator component to a simple parabola model. The example is located in the ModelCenter© file RSEgen_simple_example.pxc located in the *ProbWorks Example Problems* directory, *RSEgen Examples* subdirectory on the ProbWorks CD. This file contains the ProbWorks_RSEgenerator component connected to a script, parabola, which specifies two inputs, x1 and x2 and a single output, y1, where $y1 = x1^2+x2^2$. Figure 11 shows this example with inputs and outputs displayed from a run of the analysis. Table 18 gives the values for all input variables and Table 19 lists the values for all output variables after the analysis has been run.

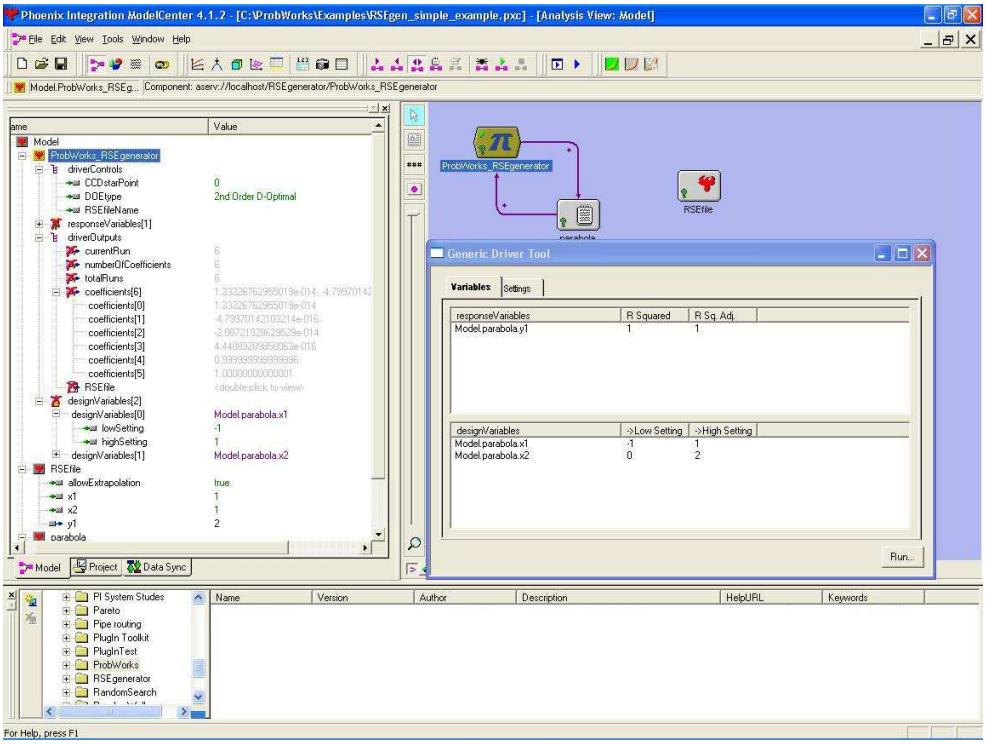


Figure 11: ProbWorks_RSEgenerator Simple Example

Table 18. ProbWorks_RSEgenerator Simple Example Inputs

VARIABLE OR SUBSECTION	VALUE
driverControls	
CCDstarPoint	0
DOEtype	2nd Order D-Optimal
RSEfileName	
responseVariables	
responseVariables[0]	Model.parabola.y1
designVariables	
designVariables[0]	Model.parabola.x1
Low Setting	-1
High Setting	1
designVariables[1]	Model.parabola.x2
Low Setting	0
High Setting	2

Table 19. ProbWorks_RSEgenerator Simple Example Outputs

VARIABLE OR SUBSECTION	VALUE
driverOutputs	
currentRun	6
numberOfCoefficients	6
totalRuns	6
coefficients	
coefficients[0]	1.33226762955019e-014
coefficients[1]	-4.79970142103214e-016
coefficients[2]	-2.08721928629529e-014
coefficients[3]	4.44089209850063e-016
coefficients[4]	0.999999999999996
coefficients[5]	1.00000000000001
RSEfile	<double-click to view>
responseVariables	
responseVariables[0]	
R Squared	1
R Sq. Adj.	1

The RSE generator created an RSE which nearly exactly matches the actual function. The RSE is only off by very slight (less than $2 * 10^{-14}$) numerical error as indicated by the R^2 value of 1. To use the resulting RSE, it can be drag and dropped into another model, or right click on the component in the example file and select "Reconnect" to update its values based on the RSE which was generated.

Complex Analysis: Analytical Function

This example demonstrates connecting the ProbWorks_RSEgenerator component to a complex discontinuous function. The example is located in the ModelCenter© file RSEgen_complex_example.pxc located in the *ProbWorks Example Problems* directory, *RSEgen Examples* subdirectory on the ProbWorks CD. This file contains the ProbWorks_RSEgenerator component connected to a script, quadratic, which has three inputs, x1,x2, and x3 and two outputs, y1 and y2, where y1=x1+2*x2+x1*x3 and $y2=x1^2+x2^2+0.5*x3$. Figure 12 shows this example with inputs and outputs displayed from

a run of the analysis. Table 20 gives the values for all input variables and Table 21 lists the values for all output variables after the analysis has been run.

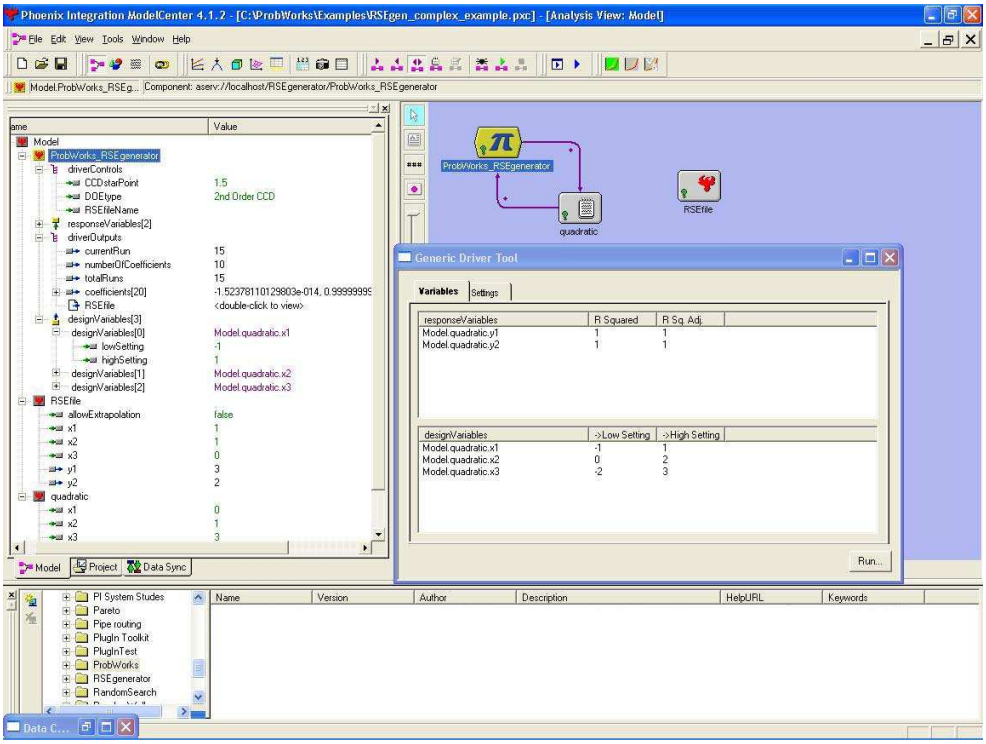


Figure 12: ProbWorks_RSEgenerator Complex Example

Table 20. ProbWorks_RSEgenerator Complex Example Inputs

VARIABLE OR SUBSECTION	VALUE
driverControls	
CCDstarPoint	1.5
DOEtype	2nd Order CCD
RSEfileName	
rseponseVariables	
responseVariables[0]	Model.quadratic.y1
responseVariables[1]	Model.quadratic.y2
designVariables	
designVariables[0]	Model.quadratic.x1
Low Setting	-1
High Setting	1
designVariables[1]	Model.quadratic.x2
Low Setting	0
High Setting	2
designVariables[2]	Model.quadratic.x3
Low Setting	-2
High Setting	3

Table 21. ProbWorks_RSEgenerator Complex Example Outputs

VARIABLE OR SUBSECTION	VALUE
driverOutputs	
currentRun	15
numberOfCoefficients	10
totalRuns	15
coefficients	-1.52378110129803e-014, 0.999999999999998, 1.99999999999998, 1.74860126378462e-015, 2.62423567176174e-015, 0.999999999999999, -9.84047912599975e-016, 8.54871728961371e-015, 1.33226762955019e-014, 3.33066907387547e-016, -2.04281036531029e-014, 8.44586840078794e-016, -1.77635683940025e-015, 0.500000000000001, 8.74964800002327e-017, -1.23581176239918e-018, 2.67939967088523e-016, 1, 1, -2.77555756156289e-017
RSEfile	<double-click to view>
responseVariables	
responseVariables[0]	
R Squared	1
R Sq. Adj.	1
responseVariables[0]	
R Squared	1
R Sq. Adj.	1

The RSE generator found nearly an exact match to this quadratic function and will find very close matches for any functions that are generally quadratic. For functions that are not fully quadratic, RSE's still may be applicable for small locations in the design space where many complex functions are near quadratic.

CHAPTER 5: PROBWORKS – PARETO ANALYSIS

OVERVIEW

The ProbWorks_ParetoAnalysis component determines the most important design variables for a model. This component allows the designer to quickly and easily assess which variables will provide the most improvement in a model or which variables should be included in later analyses such as in creating a response surface approximation or executing a Monte Carlo analysis.

MATHEMATICAL FOUNDATION

A Pareto analysis determines the impact of a set of variables on a set of responses. The ProbWorks_ParetoAnalysis determines the impact of specified primary variables on specified responses from a model. Each design variable is varied and the change in the response recorded. The variables are then ranked in importance according to how much each contributes to the variance about the mean of an output variable. Figure 13 illustrates this concept, with a set of design variables ranked and their contribution to the response graphically displayed.

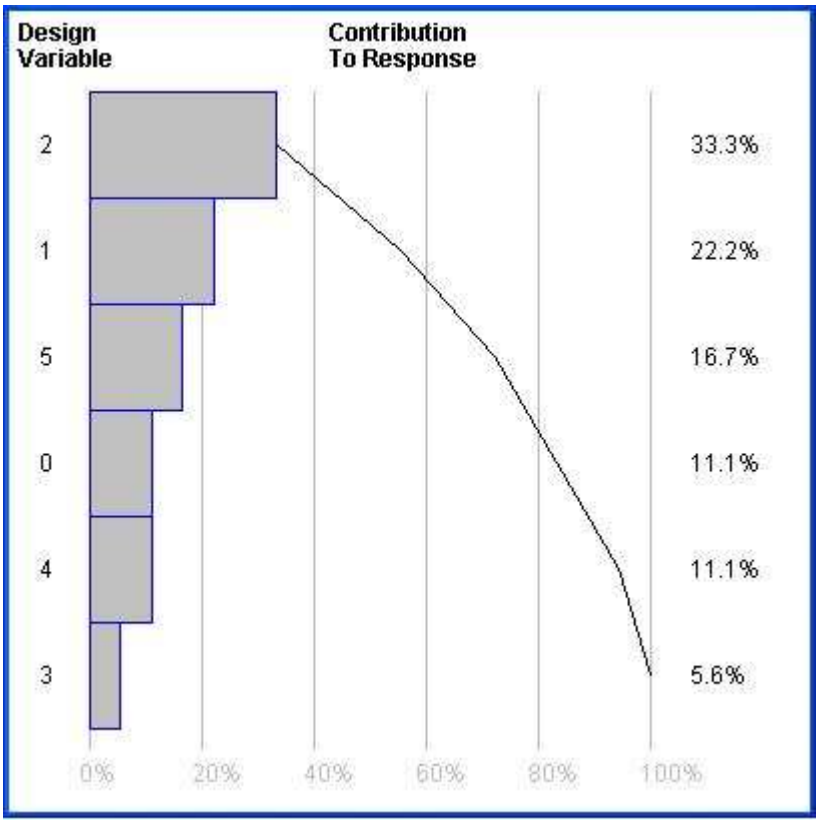


Figure 13: Example Pareto Chart

INPUT-OUTPUT VARIABLES

Inputs

Table 22 lists all of the inputs along with their type and available values.

Table 22. ProbWorks_ParetoAnalysis Inputs

VARIABLE OR SUBSECTION	TYPE	AVAILABLE VALUES
driverControls		
defaultStep	Double	≥ 0
defaultNominal	Double	Any
responseVariables	Double	Any
designVariables	Double	Any
Nominal Value	Double	Any
Step Size	Double	≥ 0

Driver Controls

These inputs are specified in the driverControls subsection of the component's variable tree in the ModelCenter© Component Tree.

- **defaultStep**: if a design variable has 0 specified as its step size, the default step size value will be used instead. This allows for many input variables to be analyzed without having to specify step sizes for all of them. All step sizes should be representative of the variables' normal ranges, or some variables could provide more or less contribution to the responses than they actual provide.
- **defaultNominal**: if a design variable has 0 specified for its step size, the default nominal value will be used (and will overwrite that variable's specified nominal value). This allows for many input variables to be analyzed without having to specify nominal values for all of them. The nominal value should be a representative point of each variable's range, though it is not as critical as the step size.

Response Variables

- Response variables are specified by dragging outputs from the model in the Component Tree into the *responseVariables* block of the Generic Driver Tool. Double-clicking on the component accesses this tool window.

Design Variables

Design variables are specified by dragging inputs from the model in the Component Tree into the *designVariables* block of the Generic Driver Tool. Double-clicking on the component accesses this tool window. Each design variable also has inputs to be specified.

- **Nominal Value**: a representative point for this design variable's range. Each step change of the design variable will occur around this point.
- **Step Size**: a representative step size for this variable. If this is left to its default value of 0, the default step size and nominal values will be used for this design variable. All step sizes for an analysis should represent similar segments of the design variable's ranges, such as 10% of each variable's normal range.

Outputs

Driver Outputs

These outputs are displayed in the driverOutputs subsection of the component's variable tree in the ModelCenter© Component Tree.

- **currentRun**: the current run the Pareto analysis is performing.
- **totalRuns**: the total number of runs to be performed.

Design Variables

These outputs are displayed in the Generic Driver Tool, which is accessed by double clicking on the Pareto analysis component. The component calculates information about each design variables impact on each response variable. The results displayed are:

- **Contribution:** the percentage contribution of this design variable to the total response for each response variable. This output is a comma separated list: the first value represents the contribution to the first response variable, the second value the contribution to the second design variable, etc.
- **Importance:** the importance ranking of each design variable with respect to each response variable, with 1 representing the most important design variable. This output is a comma separated list: the first value represents the importance with respect to the first response variable, the second value the importance with respect to the second design variable, etc.

Error Messages

Some input variable settings will cause error messages. Table 23 lists all of the error messages and what should be done to resolve the problem.

Table 23. ProbWorks_ParetoAnalysis Error Messages and Resolutions

ERROR MESSAGE	RESOLUTION
At least one designVariable must be specified: double click the ProbWorks_ParetoAnalysis component to open the Generic Driver Tool then drag input variables from other components into the designVariables box and set up the nominal value and step size	Add a design variable
At least one responseVariable must be specified: double click the ProbWorks_ParetoAnalysis component to open the Generic Driver Tool then drag output variables from other components into the responseVariables box	Add an response variable
Each designVariable must have a Step Size greater than zero or the defaultStep must be greater than zero	Either specify a non zero Step Size for each design variable or specify a non zero defaultStep
The design variables did not have any effect on at least one response variable, change the design variables' nominal values and/or ranges	Ensure that the design variables contribute to the output of each response variable

SAMPLE USAGE

Simple Analysis: Analytical Function

This example demonstrates connecting the ProbWorks_ParetoAnalysis component to a simple sum model. The example is located in the ModelCenter© file Pareto_simple_example.pxc located in the *ProbWorks Example Problems* directory, *Pareto Examples* subdirectory on the ProbWorks CD. This file contains the ProbWorks_ParetoAnalysis component connected to a script, simple_sum, which specifies three inputs, x1, x2, and x3 and a single output, y1, where y1 = x1+x2+x3. Figure 14 shows this example with inputs and outputs displayed from a run of the analysis. Table 24 gives the values for all input variables and Table 25 lists the values for all output variables after the analysis has been run.

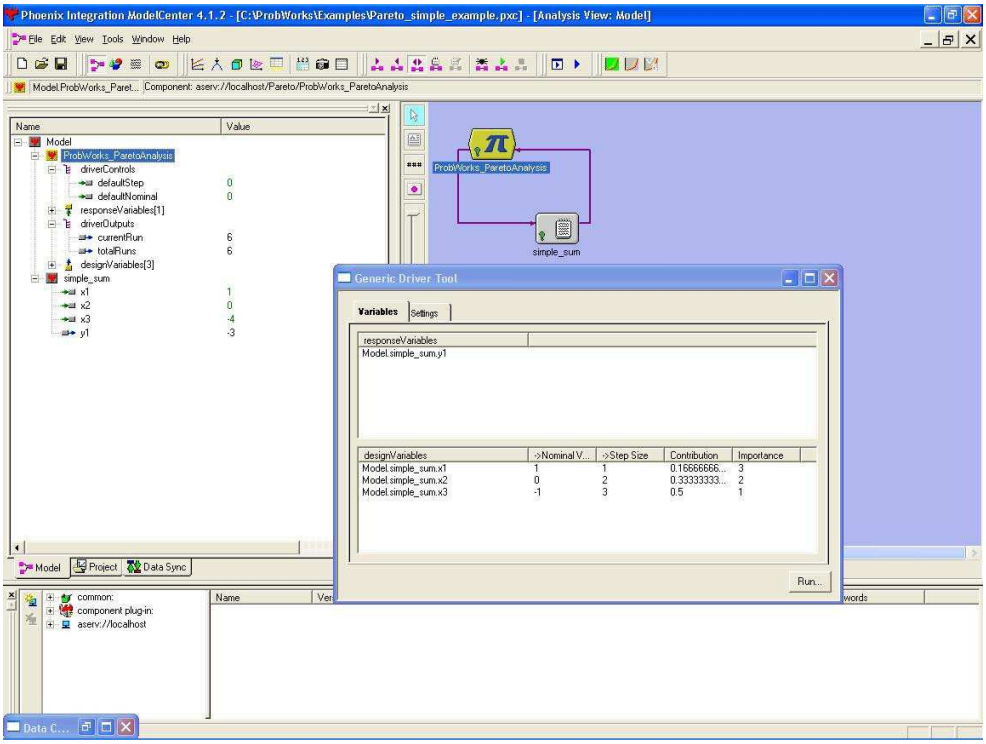


Figure 14: ProbWorks_ParetoAnalysis Simple Example

Table 24. ProbWorks_ParetoAnalysis Simple Example Inputs

VARIABLE OR SUBSECTION	VALUE
driverControls	
defaultStep	0
defaultNominal	0
responseVariables	
responseVariables[0]	Model.simple_sum.y1
designVariables	
designVariables[0]	Model.simple_sum.x1
Nominal Value	1
Step Size	1
designVariables[1]	Model.simple_sum.x2
Nominal Value	0
Step Size	2
designVariables[3]	Model.simple_sum.x3
Nominal Value	-1
Step Size	3

Table 25. ProbWorks_ParetoAnalysis Simple Example Outputs

VARIABLE OR SUBSECTION	VALUE
driverOutputs	
currentRun	6
totalRuns	6
designVariables	
designVariables[0]	
Contribution	0.1666…
Importance	3
designVariables[1]	
Contribution	0.3333…
Importance	2
designVariables[2]	
Contribution	0.5
Importance	1

Due to all the inputs having equal waiting in the equation of the single output, the importance for each design variable was determined by its step size. This example demonstrates the basic operation of the Pareto analysis component and underscores the importance of properly picking step sizes so that they do not alter the influence of a design variable.

Complex Analysis: Analytical Function

This example demonstrates connecting the ProbWorks_ParetoAnalysis component to a more complex function. The example is located in the ModelCenter© file Pareto_complex_example.pxc located in the *ProbWorks Example Problems* directory, *Pareto Examples* subdirectory on the ProbWorks CD. This file contains the ProbWorks_ParetoAnalysis component connected to a script, complex_sum, which has six inputs, x1, x2, x3, x4, x5, x6 and two outputs, y1 and y2, where y1=x1+2*x2+3*x3+0.5*x4+x5+1.5*x6 and y2=3*x1+x2+2*x3+x4+0*x5+0.5*x6. Figure 15 shows this example with inputs and outputs displayed from a run of the analysis. Table 26 gives the values for all input variables and Table 27 lists the values for all output variables after the analysis has been run.

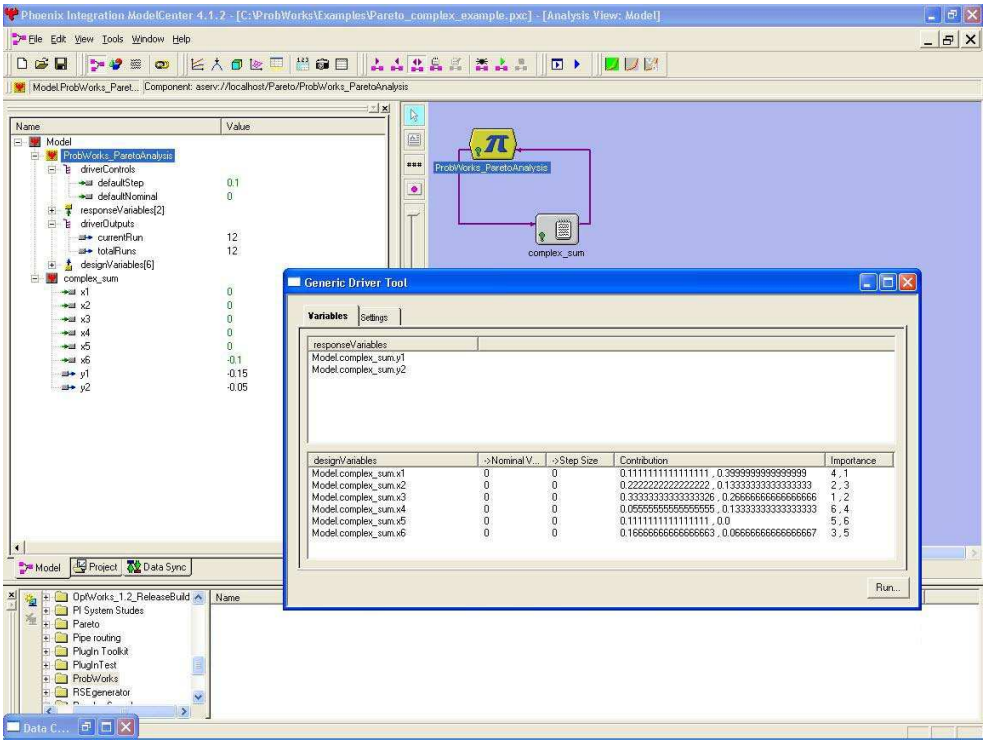


Figure 15: ProbWorks_ParetoAnalysis Complex Example

Table 26. ProbWorks_ParetoAnalysis Complex Example Inputs

VARIABLE OR SUBSECTION	VALUE
driverControls	
defaultStep	0.1
defaultNominal	0
responseVariables	
responseVariables[0]	Model.complex_sum.y1
responseVariables[1]	Model. complex_sum.y2
designVariables	
designVariables[0]	Model. complex_sum.x1
Nominal Value	0
Step Size	0
designVariables[1]	Model. complex_sum.x2
Nominal Value	0
Step Size	0
designVariables[2]	Model. complex_sum.x3
Nominal Value	0
Step Size	0
designVariables[3]	Model. complex_sum.x4
Nominal Value	0
Step Size	0
designVariables[4]	Model. complex_sum.x5
Nominal Value	0
Step Size	0
designVariables[5]	Model. complex_sum.x6
Nominal Value	0
Step Size	0

Table 27. ProbWorks_ParetoAnalysis Complex Example Outputs

VARIABLE OR SUBSECTION	VALUE
driverOutputs	
currentRun	12
totalRuns	12
designVariables	
designVariables[0]	
Contribution	0.1111…, 0.3999…
Importance	4,1
designVariables[1]	
Contribution	0.2222…, 0.1333…
Importance	2,3
designVariables[2]	
Contribution	0.3333…, 0.2666…
Importance	1,2
designVariables[3]	
Contribution	0.0555…, 0.1333…
Importance	6,4
designVariables[4]	
Contribution	0.1111…, 0.0
Importance	5,6
designVariables[5]	
Contribution	0.1666…, 0.0666…
Importance	3,5

Each design variable's importance was determined based on its coefficient in the complex_sum script, and each variable had a different importance for the two response variables.

CHAPTER 6: PROBWORKS – GRAPH PLUG-IN

OVERVIEW

The ProbWorks_GraphPlugIn is not an analysis component such as the other components of the ProbWorks suite, but rather a plug-in utility designed to provide fast-acting graphical analysis of the results from other analyses. Specifically, the graph plug-in produces histograms from Monte Carlo runs and Pareto charts from sensitivity analyses. Its installation (covered in Chapter 1) and operation are different from the other analysis components. The operation of the component will be detailed and two examples given in this chapter.

COMPONENT OPERATION

Once the ProbWorks_GraphPlugIn has been installed, it will be available from the Data Collector after any analysis is performed in ModelCenter©. Figure 16 indicates the graph plug-in icon in the Data Collector after a run of the Pareto analysis. Clicking on this icon runs the utility, as shown in Figure 17.

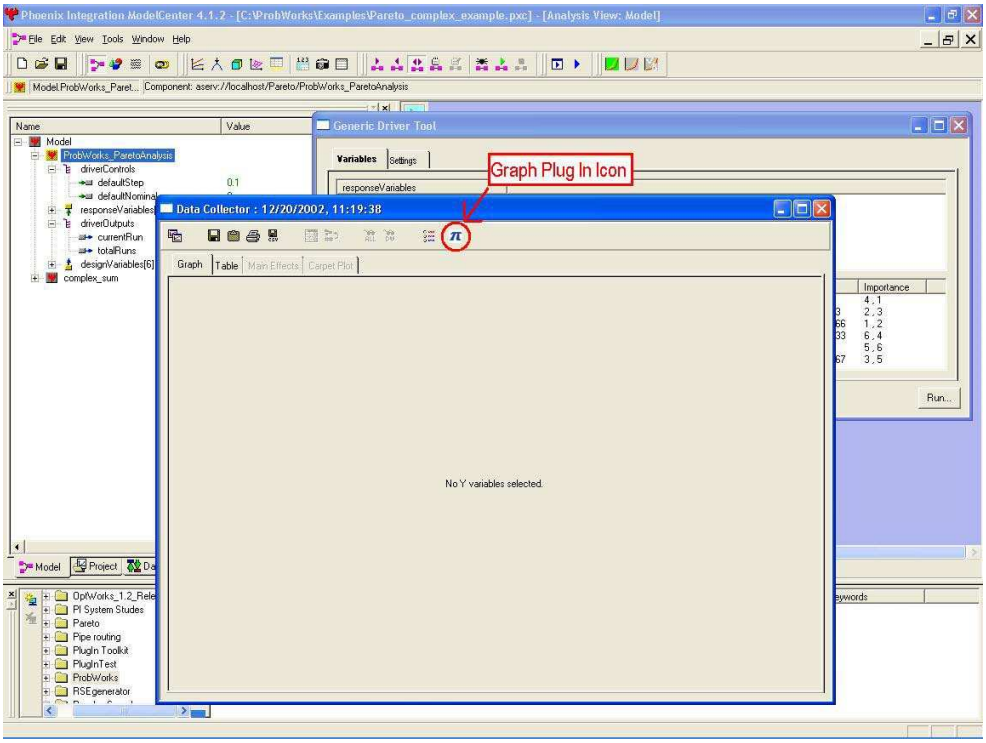


Figure 16: Graph Plug-in Icon

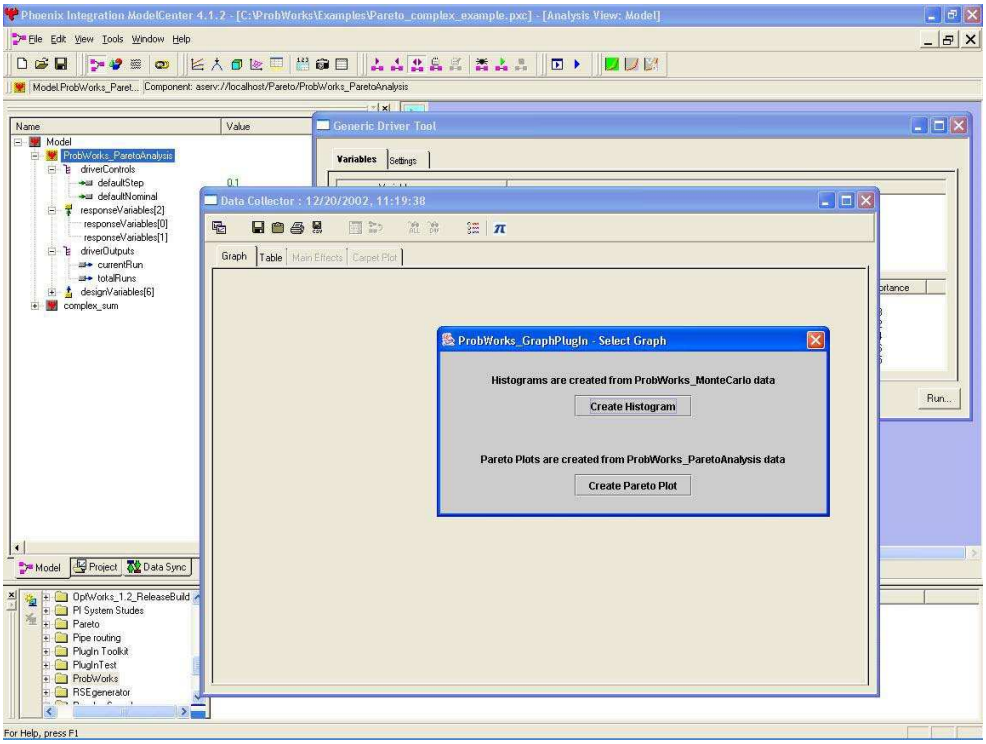


Figure 17: Graph Plug-in Initial Interface

From this first interface, the user can select to create either a histogram from Monte Carlo data or a Pareto chart from a Pareto analysis. Figures 18 and 19 illustrate the set of menus which appear for each of these choices. The user can now specify which variables to graph by either typing the complete variable name into the text box, or dragging the variable from the Component Tree into the text box and then clicking the Create Histogram or Create Pareto Plot button. Histograms can be created from nearly any numerical variable within ModelCenter©, such as the x1variable from a simple script: Model.pass_through.x1. Pareto charts can only be created from individual response variables in a ProbWorks_ParetoAnalysis component.

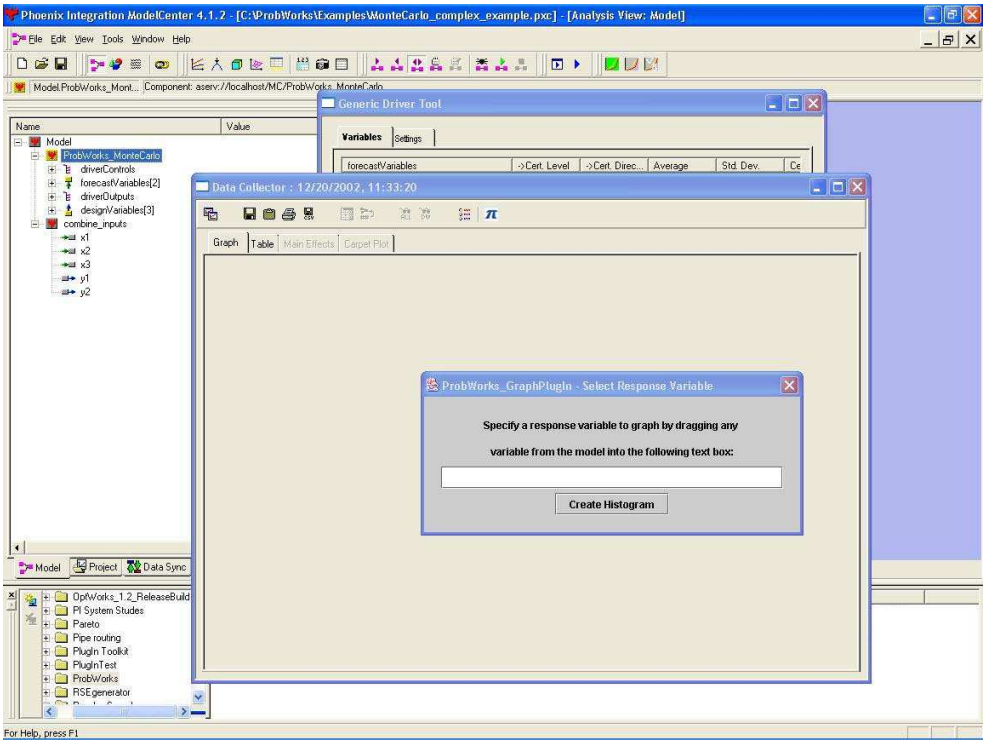

Phoenix Integration ModelCenter 4.1.2 - [C:\ProbWorks\Examples\MonteCarlo_complex_example.pxc] - [Analysis View: Model]
File Edit View Tools Window Help
Generic Driver Tool
Variables Settings
Data Collector : 12/20/2002, 11:33:20
Graph Table Main Effects Carpet Plot
ProbWorks_GraphPlugIn - Select Response Variable
Specify a response variable to graph by dragging any
variable from the model into the following text box:
Create Histogram
Model Project Data Sync
For Help, press F1

Figure 18: Graph Plug-in Histogram Interface

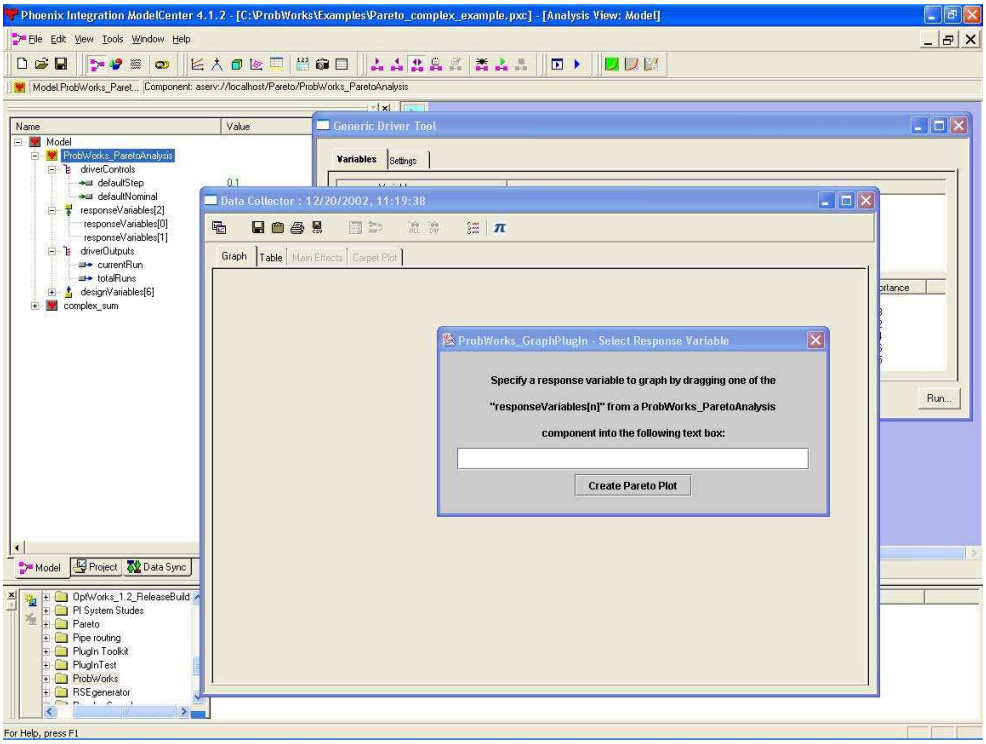

Phoenix Integration ModelCenter 4.1.2 - [C:\ProbWorks\Examples\Pareto_complex_example.pxc] - [Analysis View: Model]
File Edit View Tools Window Help
Generic Driver Tool
Variables Settings
Data Collector : 12/20/2002, 11:19:38
Graph Table Main Effects Carpet Plot
ProbWorks_GraphPlugIn - Select Response Variable
Specify a response variable to graph by dragging one of the
"responseVariables[n]" from a ProbWorks_ParetoAnalysis
component into the following text box:
Create Pareto Plot
Model Project Data Sync
For Help, press F1

Figure 19: Graph Plug-in Pareto Chart Interface

SAMPLE USAGE

Monte Carlo Analysis: Histogram

This example demonstrates graphing histograms from the output of a ProbWorks_MonteCarlo analysis with the ProbWorks_GraphPlugIn utility. The example used is the complex Monte Carlo example used in Chapter 2 and is the MonteCarlo_complex_example.pxc file located in the *ProbWorks Example Problems* directory, *Monte Carlo Examples* subdirectory on ProbWorks CD. Open up this model, double click the Monte Carlo component to bring up the Generic Driver Tool, and click the Run button to start the Monte Carlo analysis with the Data Collector. Once the model has finished running, click the Graph Plug-in icon at the top of the Data Collector to run the Graph Plug-in Utility. Select the Create Histogram button. Finally, drag the output variable y2 from the Component Tree under the combine_inputs script into the text box on the Graph Plug-in and click the Create Histogram button to create a histogram of this response. Figure 20 shows the Graph Plug-in with the y2 variable selected and the resulting graph.

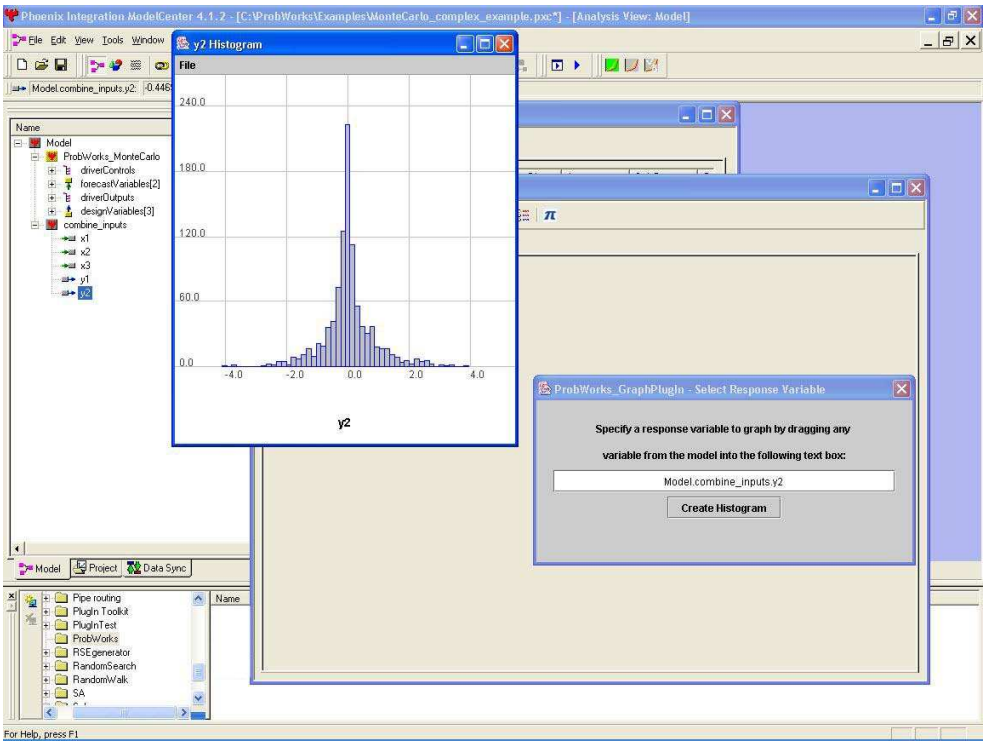


Figure 20: Graph Plug-in Histogram Example

The label is initially the name of the variable used to create the graph, in this case y2, but the label can be selected and a more complete name typed in. The user may also select "Save as jpg" from the File menu to save the graph as an image file. The histogram shows the distribution of the output variable as the design variables were varied. For each range of

values in the histogram, the total number of times that value was recorded is displayed. The high peak in the middle shows the concentration of values around this distribution's mean.

Pareto Analysis: Pareto Chart

This example demonstrates graphing Pareto charts from the output of a ProbWorks_ParetoAnalysis analysis with the ProbWorks_GraphPlugIn utility. The example used is the complex Pareto analysis example used in Chapter 5 and is the Pareto_complex_example.pxc file located in the *ProbWorks Example Problems* directory, *Pareto Examples* subdirectory on ProbWorks CD. Open up this model, double click the Pareto analysis component to bring up the Generic Driver Tool, and click the Run button to start the Pareto analysis with the Data Collector. Once the model has finished running, click the Graph Plug-in icon at the top of the Data Collector to run the Graph Plug-in Utility. Select the Create Pareto Plot button. Finally, drag the response variable **responseVariables[0**] from the Component Tree under the ProbWorks_ParetoAnalysis component, **responseVariables[2]** group into the text box on the Graph Plug-in and click the Create Pareto Plot button to create a Pareto chart of this response. Figure 21 shows the Graph Plug-in with the first response variable selected and the resulting graph.

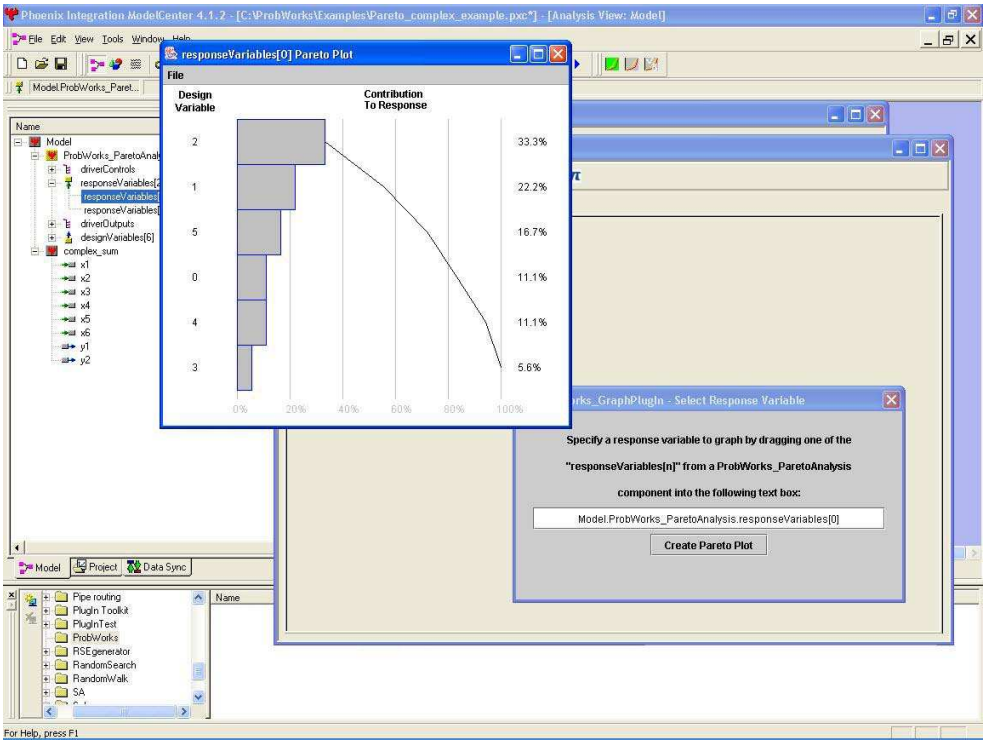


Figure 21: Graph Plug-in Pareto Chart Example

The Pareto chart lists the design variables (starting from 0, the index ModelCenter© starts with) in the order of importance. The labels are initially the numbers of the design variables, but the labels can be selected and more complete names typed in. The user may also select “Save as .jpg” from the File menu to save the graph as an image file. The bar graph portion displays the contribution of each variable to the response, with a line indicating the cumulative response of the variables. The exact percent contribution is also given to the right of each variable.

CHAPTER 7: PROBWORKS – COMPLETE PROBABILISTIC PROCESS

OVERVIEW

The ProbWorks components can not only assist individually in performing probabilistic analyses, but together encapsulate an entire probabilistic process, allowing the analysis of many different types of models. This chapter shows the use of each component within a process for probabilistic analysis and provides an example of this process.

PROBABILISTIC ANALYSIS PROCESS

A probabilistic analysis involves applying distributions to the inputs of a model and creating distributions for the resulting outputs. These resulting distributions can provide information about uncertainty due to the environment, the analysis process, or the model itself. A Monte Carlo analysis provides a straightforward, accurate way to apply the distributions and determine their effects; however, it requires thousands of separate analyses of the model. As the model's run time increases, the time required for a Monte Carlo analysis can increase to infeasible levels. Tools, such as the DPOMD method, exist that provide approximations to a Monte Carlo analysis and can be used when appropriate to the model. In addition, approximations can be made to the model itself or individual components in a model, such as creating RSE's. A Pareto analysis can help determine the most appropriate variables for a DPOMD analysis or to fit with an RSE for models with particularly long run times and many variables. The key to a successful probabilistic analysis is in choosing the right methods for a particular model. Figure 22 provides a basic flowchart of this process and the next section covers and example using the ProbWorks components.

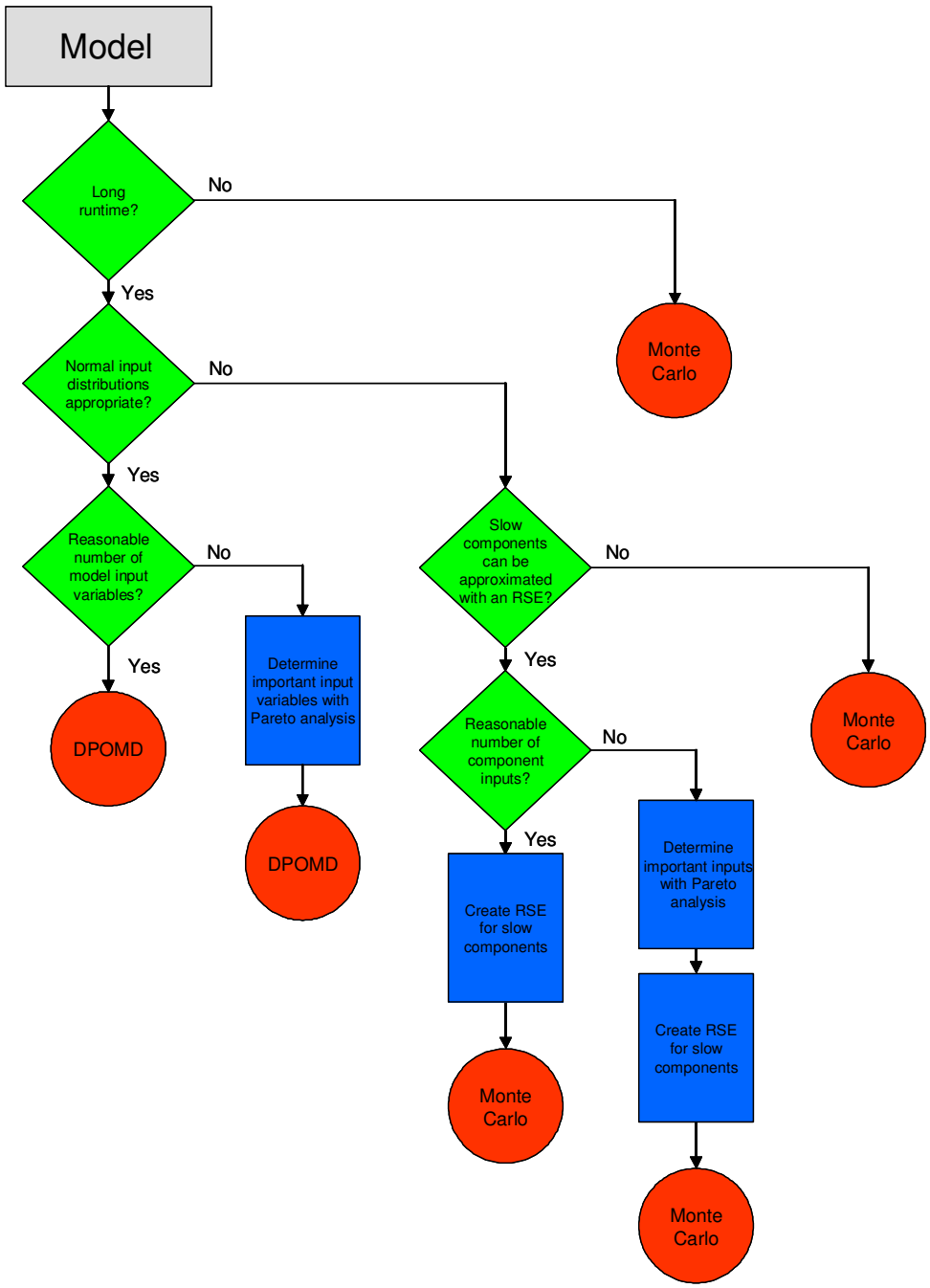


Figure 22: Probabilistic Analysis Flowchart

PROBABILISTIC ANALYSIS EXAMPLE

This example demonstrates using all of the ProbWorks components together for a complete probabilistic assessment. The example file is called process_example.pxc and is located in the *ProbWorks Examples* directory, *Generic Example Scripts Used* subdirectory. In this example, 3 analysis components are represented by the simple script components component1, component2, and component3. In most cases, the analysis components would

be complex programs that could take a long time to perform each analysis. For this example, component2 represents an analysis component with a long run time and therefore an RSE is created for use in the uncertainty analysis. Because the number of runs necessary to create an RSE can scale exponentially with the number of input variables, the first step is to perform a Pareto analysis on the component and determine the most important input variables. Input variables 1 through 6 are the most important, so they are then used to create an approximation to component2 with the RSE generator. The newly created RSE is linked with the other two analysis components and both the Monte Carlo component and the DPOMD component are connected to the model. Normally only one or the other would be used, depending on the run time of the total model. In this example case, both are used to show their relative features. The Monte Carlo allows for a wide variety of input distributions but takes far longer to run. The DPOMD analysis produces similar results, but with far fewer runs. The results are different primary due to the use of different distribution types. Figure 23 shows the model file with all of the ProbWorks components.

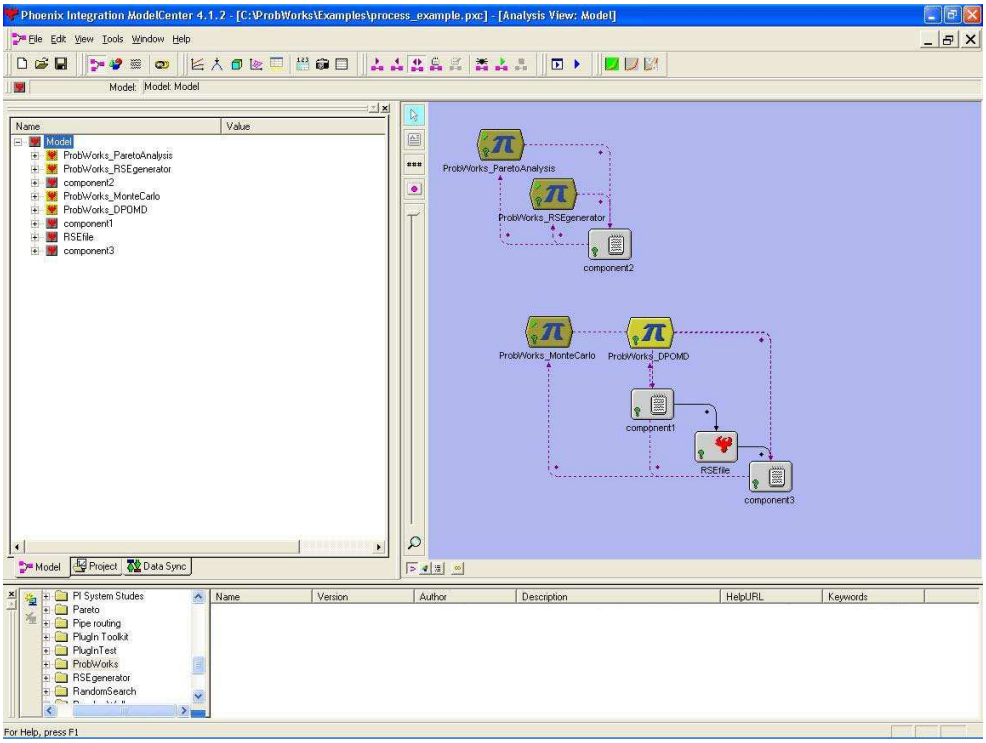


Figure 23: Probabilistic Analysis Example

REFERENCE LIST

1. Hayter, Anthony J., “Probability and Statistics for Engineers and Scientists,” PWS Publishing Co., Boston, MA, 1996.
2. Weisstein, Eric, “Eric Weisstein’s World of Mathematics,” Wolfram Research, http://mathworld.wolfram.com/, last accessed: December 2002.
3. McCormick, D., Olds, J., "Approximation of Probabilistic Distributions Using Selected Discrete Simulations," AIAA 2000-4863, 8th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, Long Beach, CA, September 6-8, 2000.
4. McCormick, D., “Distributed Uncertainty Analysis Techniques for Conceptual Launch Vehicle Design,” Doctoral Thesis, Georgia Institute of Technology, 2001.

LIST OF ACRONYMS

CCD	Central Composite Design
CD	Compact Disc
DPOMD	Discrete Probability Optimal Matching Distributions
DOE	Design of Experiments
FF	Fractional Factorial
PC	Personal Computer
RSE	Response Surface Equation

LIST OF FIGURES

Figure 1: Monte Carlo Component Interface Elements _____ 24
Figure 2: Monte Carlo Component Sample Error Message _____ 25
Figure 3: Sample Monte Carlo Setup _____ 27
Figure 4: Example Monte Carlo Distribution _____ 32
Figure 5: ProbWorks_MonteCarlo Simple Example _____ 38
Figure 6: ProbWorks_MonteCarlo Complex Example _____ 40
Figure 7: Illustration of Discrete Distribution Derived from Continuous Distribution[4] _____ 44
Figure 8: ProbWorks_DPOMD Simple Example _____ 48
Figure 9: ProbWorks_DPOMD Complex Example _____ 50
Figure 10: Example Response Surface Equation _____ 54
Figure 11: ProbWorks_RSEgenerator Simple Example _____ 58
Figure 12: ProbWorks_RSEgenerator Complex Example _____ 61
Figure 13: Example Pareto Chart _____ 65
Figure 14: ProbWorks_ParetoAnalysis Simple Example _____ 69
Figure 15: ProbWorks_ParetoAnalysis Complex Example _____ 72
Figure 16: Graph Plug-in Icon _____ 75
Figure 17: Graph Plug-in Initial Interface _____ 76
Figure 18: Graph Plug-in Histogram Interface _____ 77
Figure 19: Graph Plug-in Pareto Chart Interface _____ 77
Figure 20: Graph Plug-in Histogram Example _____ 78
Figure 21: Graph Plug-in Pareto Chart Example _____ 79
Figure 22: Probabilistic Analysis Flowchart _____ 82
Figure 23: Probabilistic Analysis Example _____ 83

LIST OF TABLES

Table 1. Overview of ProbWorks ModelCenter© Components ____ 15
Table 2. ProbWorks Installation CD Overview ____ 18
Table 3. ProbWorks_MonteCarlo Inputs ____ 33
Table 4. Parameters to Define Various Distributions ____ 36
Table 5. ProbWorks_MonteCarlo Error Messages and Resolutions ____ 37
Table 6. ProbWorks_MonteCarlo Simple Example Inputs ____ 39
Table 7. ProbWorks_MonteCarlo Simple Example Outputs ____ 39
Table 8. ProbWorks_MonteCarlo Complex Example Inputs ____ 41
Table 9. ProbWorks_MonteCarlo Complex Example Outputs ____ 42
Table 10. ProbWorks_DPOMD Inputs ____ 45
Table 11. ProbWorks_DPOMD Error Messages and Resolutions ____ 47
Table 12. ProbWorks_DPOMD Simple Example Inputs ____ 49
Table 13. ProbWorks_DPOMD Simple Example Outputs ____ 49
Table 14. ProbWorks_DPOMD Complex Example Inputs ____ 51
Table 15. ProbWorks_DPOMD Complex Example Outputs ____ 52
Table 16. ProbWorks_RSEgenerator Inputs ____ 54
Table 17. ProbWorks_RSEgenerator Error Messages and Resolutions ____ 57
Table 18. ProbWorks_RSEgenerator Simple Example Inputs ____ 59
Table 19. ProbWorks_RSEgenerator Simple Example Outputs ____ 60
Table 20. ProbWorks_RSEgenerator Complex Example Inputs ____ 62
Table 21. ProbWorks_RSEgenerator Complex Example Outputs ____ 63
Table 22. ProbWorks_ParetoAnalysis Inputs ____ 66
Table 23. ProbWorks_ParetoAnalysis Error Messages and Resolutions ____ 68
Table 24. ProbWorks_ParetoAnalysis Simple Example Inputs ____ 70
Table 25. ProbWorks_ParetoAnalysis Simple Example Outputs ____ 71
Table 26. ProbWorks_ParetoAnalysis Complex Example Inputs ____ 73
Table 27. ProbWorks_ParetoAnalysis Complex Example Outputs ____ 74

INDEX

A

allowExtrapolation
input in ProbWorks_RSEgenerator, 56
Analysis View
window in ModelCenter environment, 24
Average
output in ProbWorks_MonteCarlo, 36

B

Beta
(see Parameter 2)
Beta Distribution
definition, 35

C

CCDstarPoint
input in ProbWorks_RSEgenerator, 55
Cert. Direction
input in ProbWorks_DPOMD, 46
input in ProbWorks_MonteCarlo, 34
Cert. Level
input in ProbWorks_DPOMD, 46
input in ProbWorks_MonteCarlo, 33
Cert. Value
output in ProbWorks_DPOMD, 47
output in ProbWorks_MonteCarlo, 36
Certainty Values, 31, 44
coefficients
output ProbWorks_RSEgenerator, 56
Component Tree
window in ModelCenter environment, 24
Contribution
output in ProbWorks_ParetoAnalysis, 67
covariance
input in ProbWorks_DPOMD, 45
currentRun
output in ProbWorks_DPOMD, 46
output in ProbWorks_ParetoAnalysis, 67
output in ProbWorks_RSEgenerator, 55
currentTrial
output in ProbWorks_MonteCarlo, 36

D

defaultNominal

input in ProbWorks_ParetoAnalysis, 66
defaultStep
input in ProbWorks_ParetoAnalysis, 66
designVariables
input in ProbWorks_DPOMD, 46
input in ProbWorks_MonteCarlo, 34
input in ProbWorks_ParetoAnalysis, 67
input in ProbWorks_RSEgenerator, 55
Distribution
implementation in probabilistic process, 81
plotting with ProbWorks_GraphPlugIn, 78
usage in ProbWorks_DPOMD, 43-44
usage in ProbWorks_MonteCarlo, 31-32, 34, 36
DOEtype
input in ProbWorks_RSEgenerator, 55
driverControls
input group in ProbWorks_DPOMD, 45
input group in ProbWorks_MonteCarlo, 33
input group in ProbWorks_ParetoAnalysis, 66
input group in ProbWorks_RSEgenerator, 55

E

Exponential Distribution
definition, 34

F

Forecast Variable
general usage, 24
in ProbWorks_DPOMD, 43
in ProbWorks_MonteCarlo, 31
forecastVariables
input section in ProbWorks_DPOMD, 46
input section in ProbWorks_MonteCarlo, 33
output section in ProbWorks_DPOMD, 47
output section in ProbWorks_MonteCarlo, 36

G

Generic Driver Tool
window in ModelCenter environment, 24

H

Hasofer-Lind Transform
application to DPOMD, 43
High Setting
input in ProbWorks_RSEgenerator, 55

I

Importance
output in ProbWorks_ParetoAnalysis, 67
Installation, 25-26

J

Java program, 21

K

Kurtosis
output in ProbWorks_ModleCenter, 36

I

Lognormal Distribution
definition, 35
Low Setting
input in ProbWorks_RSEgenerator, 55

M

Mean
input in ProbWorks_DPOMD, 46
output in ProbWorks_DPOMD, 47

N

Nominal Value
input in ProbWorks_ParetoAnalysis, 67
Normal Distribution
definition, 34
numberOfCoefficients
output in ProbWorks_RSEgenerator, 55
numberOfTrials
input in ProbWorks_MonteCarlo, 33

P

Parameter 1
input in ProbWorks_MonteCarlo, 35
Parameter 2
input in ProbWorks_MonteCarlo, 35
Parameter 3
input in ProbWorks_MonteCarlo, 35

R

R Sq. Adj.
output in ProbWorks_RSEgenerator, 56
R squared
output in ProbWorks_RSEgenerator, 56

reduction
input in ProbWorks_DPOMD, 45
responseVariables
input in ProbWorks_ParetoAnalysis. 66
input in ProbWorks_RSEgnerator, 55
output in ProbWorks_RSEgenerator, 56
RSEfile
output in ProbWorks_RSEgenerator, 56
RSEfileName
input in ProbWorks_RSEgenerator, 55

S

Server Browser
window in ModelCenter environment, 24
Skewness
output in ProbWorks_MonteCarlo, 36
Std. Dev.
input in ProbWorks_DPOMD, 46
output in ProbWorks_DPOMD, 47
output in ProbWorks_MonteCarlo, 36
Step Size
input in ProbWorks_ParetoAnalysis, 67

T

totalRuns
output in ProbWorks_DPOMD, 46
output in ProbWorks_ParetoAnalysis, 67
output in ProbWorks_RSEgenerator, 56
Triangle Distribution
definition, 34

U

Uniform Distribution
definition, 34

W

Weibull Distribution
definition, 34

PRODUCTION NOTES

This manual was created electronically using Microsoft Word and Adobe Acrobat. Graphic art was produced using Adobe Photoshop and Denaba Canvas. Times Roman, Arial Narrow, and Arial black typefaces are used throughout this book.